ORSAY
N° d'ordre : 3348

UNIVERSITE DE PARIS-SUD

CENTRE D'ORSAY

THESE

PRESENTEE

POUR OBTENIR

LE TITRE DE DOCTEUR 3eme CYCLE

SPECIALITE : ELECTRONIQUE

PAR

RAVELOSON Christian Hary Mala

Sujet de la thèse

ARCHITECTURE DE SYSTEME GRAPHIQUE A BALAYAGE DE TRAME POUR LA SYNTHESE D'IMAGES EN VUE DE L'APPLICATION DANS LE DOMAINE DE LA C.A.O.

Soutenue le 6 Décembre 1982 devant le jury composé de

Messieurs	*M.*	*DESAINTFUSCIEN*	*Président*
	B.	*GRABOWSKI*	*Examinateurs*
	R.J.	*PISTENON*	
	F.	*HORUSITZKY*	
	C.	*BELLEVAUX*	

ORSAY
N° d'ordre : 3348

UNIVERSITE DE PARIS-SUD

CENTRE D'ORSAY

THESE

PRESENTEE

POUR OBTENIR

LE TITRE DE DOCTEUR 3eme CYCLE

SPECIALITE : ELECTRONIQUE

PAR

RAVELOSON Christian Hary Mala

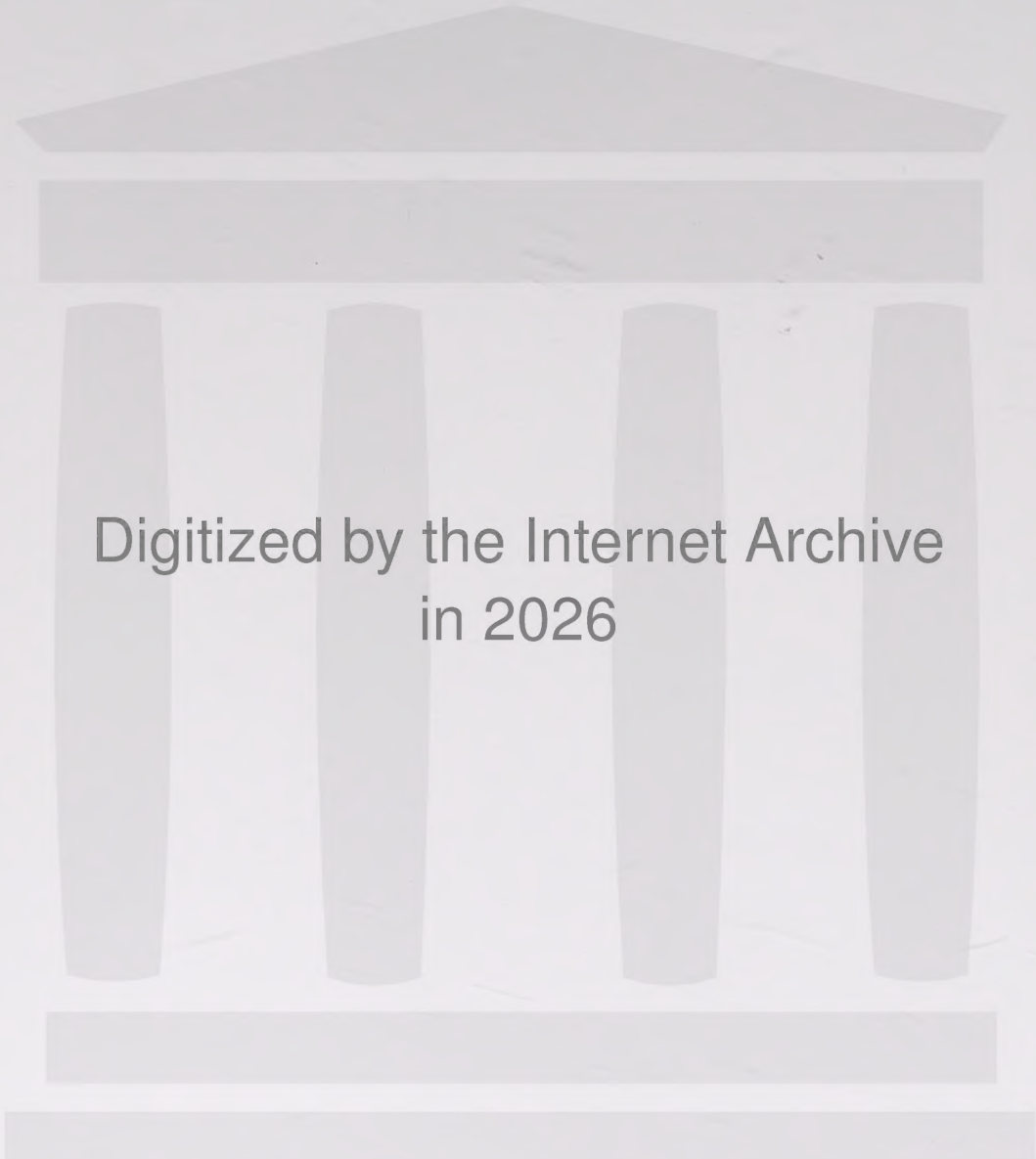
Sujet de la thèse

ARCHITECTURE DE SYSTEME GRAPHIQUE A BALAYAGE DE TRAME POUR LA SYNTHESE D'IMAGES EN VUE DE L'APPLICATION DANS LE DOMAINE DE LA C.A.O.

Soutenue le 6 Décembre 1982 devant le jury composé de

Messieurs	*M.*	*DESAINTFUSCIEN*	*Président*
	B.	*GRABOWSKI*	*Examinateurs*
	R.J.	*PISTENON*	
	F.	*HORUSITZKY*	
	C.	*BELLEVAUX*	

à ma femme
à mes parents
à mes grands-parents
à mon oncle Maurice
à ma sœur Loulou
à mon frère Riri


Digitized by the Internet Archive
in 2026

https://archive.org/details/IA41547326_0031

R E M E R C I E M E N T S

Je tiens à remercier :

Monsieur Le Professeur M. DESAINTFUSCIEN qui m'a fait l'honneur d' accepter la présidence du Jury .

Monsieur Le Professeur B. GRABOWSKI qui a suivi avec intérêt le déroulement de ce travail .

Monsieur PISTENON Directeur du Centre CPAO qui a toujours porté une attention particulière sur ce travail . Pour sa patience et ses conseils constructifs je tiens à lui exprimer ma très vive gratitude .

Monsieur F. HORUSITZKY Ingénieur au Centre CPAO dont les conseils et les encouragements ont étés particulièrement importants tout au long de l'élaborration de ce travail . Je tiens à lui exprimer toute ma reconnaissance .

Monsieur C. BELLEVAUX Chef du Groupe GII pour l'intérêt qu'il porté à ce travail en acceptant de siéger à ce Jury .

Et tout le personnel du Groupe GII .

Ainsi qu'à Monsieur R. BOUVIGNIES du Centre d'Edition de l'ENSTA pour la réalisation de ce document .

Abstract:

This thesis presents the multiprocessor architecture of a raster scan graphic display system for applications in CAD. The main part of the work is essentially the study of the display unit,in which is included the depth computation function of a Z-Buffer.

In order to speed up the algorithms,they are computed by "n" processors working in parallel. To avoid slowing these "n" processors and to have the largest bandwidth for updating the frame buffer,this one is shared into "p" independent areas.

The interconnexion between the "n" processors and "p" memory units uses an "a^n x b^n" delta network;in our case,in order to implement a "2^k x 2^r" interconnection network,we have realized a "2 x 2" crossbar switch.

Keywords:

multiprocessor architecture , raster scan , graphic display system , display unit , frame buffer , delta network , crossbar switch , interconnection network .

P L A N

0. Introduction

I. Principes généraux des systèmes graphiques à balayage de trame

1. Notions et techniques de générations d'images vidéo

1.1. Définition
1.2. Génération de l'image vidéo
1.3. Techniques de visualisation de l'image vidéo
1.4. Architecture de base d'un système graphique à balayage de trame.

2. Description d'une mémoire d'image

2.1. Instructions d'écriture et lecture en mémoire
2.2. Mode d'adressage d'une mémoire d'image

3. Les différents types de réalisation de systèmes graphiques à balayage de trame existants

3.1. Système graphique avec processeur graphique
3.2. Système graphique cablé
3.3. Système graphique simple

II. Présentation succincte des différentes composantes

1. Système graphique à mémoire de trame

2. Utilisations de systèmes à mémoire de trame

2.1. Utilisation en traitement d'image
2.2. Utilisation en synthèse d'image

3. Visualisation réaliste d'objects tridimensionnels

3.1. Modèle de représentation
3.2. Opérations de visualisation nécessaires
3.3. Remarque sur la visualisation réaliste d'objets tridimensionnels
3.4. Elimination des parties cachées

4. Unité d'affichage

4.1. Accélération des algorithmes.
4.2. Etude d' l'Unité d'affichage

III. Description des différents organes -

1. Présentation et caractéristique de l'unité d'affichage

1.1. Mode de travail image
1.2. Mode de travail espace

2. Organisation de l'unité d'affichage

3. Organe de mise à jour de l'image

3.1. Principe de fonctionnement
3.2. Organisation de la mémoire d'image
3.3. Circuit d'aiguillage
3.4. Description du circuit de mise à jour de l'image
3.5. Description du séquenceur

4. Organe vidéo

4.1. Description du circuit de rafraichissement de l'image
4.2. Description du générateur de signaux de synchronisation.

IV. Synthèse des différents organes -

1. Conception du circuit d'aiguillage

1.1. Conception des tampons
1.2. Conception du réseau d'interconnexion
1.3. Conception du circuit de décodage
1.4. Circuit d'écriture du tampon émetteur

2. Conception du circuit de mise à jour de l'image

2.1. Circuit d'adressage
2.2. Circuit de chargement du nouveau point
2.3. Circuit d'état de la zone
2.4. Circuit de chargement de l'ancien point
2.5. Circuit de comparaison
2.6. Circuit de visualisation de l'image et de rafraichissement des mémoires
2.7. Séquenceur

3. Conception du circuit de rafraichissement de l'image

4. Conception du générateur de signaux de synchronisation

V. Conclusions

VI. Bibliographie

VII. Annexe : Configuration de test.

CHAPITRE 0

INTRODUCTION

La Conception Assistée par Ordinateur est l'un des domaines privilégiés de l'utilisation de l'image : l'esquisse, le dessin, le schéma, le plan, la maquette, etc. autant de moyens pour concrétiser une idée, aider à la réflexion, communiquer entre individus ou entre l'individu et la machine.

Le poncif classique de la Visualisation Graphique par Ordinateur comme outil parfait pour réaliser une symbiose entre l'Homme et la Machine, prend tout son sens dans le cas de terminaux à balayage de trame puisque dans l'idéal, avec ce type de terminaux et avec lieu sûr, les programmes adaptés, il n'y a plus de différence entre l'image d'un objet virtuel synthétisé par ordinateur et celle de ce même objet réel filmé par une caméra de télévision.

Cependant, dans un contexte de C.A.O., ces avantages potentiels peuvent se déprécier de par une génération par trop lente des images. Dans le cas d'un modeleur tridimensionnel, dont l'objet est l'aide à la création de formes volumiques, le succès d'un tel système dépend en particulier de la rapidité et de la qualité des rendus : construction et compréhension de la forme en cours de conception sont des opérations suffisamment compliquées pouvant aboutir à une annihilation complète de l'esprit de création.

Parmi les points pénalisants de la synthèse d'un objet volumique sur un écran à mémoire de trame, notons essentiellement :

- tous les calculs d'élimination des parties cachées, que le modèle informatique de l'objet soit t polyédrique ou non,

- les calculs de discrétisation, consistant à déterminer pour chaque face de l'objet l'ensemble des points (ou "pixels") qui devront être allumés sur l'écran,

- les calculs d'éclairement (calcul de l'énergie lumineuse émise par tout point de l'objet).

Les exigences de qualité et de rapidité des rendus posent des problèmes tant au niveau des matériels que des logiciels dont la résolution est non triviale.

Nous allons présenter une architecture d'un terminal graphique à mémoire de trame permettant une parallélisation des différents calculs évoqués précédemment et prenant en charge, directement au niveau de l'unité d'affichage, les calculs d'élimination de parties cachées.

CHAPITRE I

PRINCIPES GENERAUX DES SYSTEMES GRAPHIQUES

A BALAYAGE DE TRAME

CHAPITRE DIDACTIQUE DONNANT LA DESCRIPTION DES PRINCIPALES NOTATIONS TECHNOLOGIQUES RELATIVES A L'ARCHITECTURE DES MATERIELS GRAPHIQUES A BALAYAGE DE TRAME.

PEUT NE PAS ÊTRE LU DU SPECIALISTE.

PRINCIPES GENERAUX D'UN SYSTEME GRAPHIQUE A BALAYAGE DE TRAME

1. TECHNOLOGIES DE GENERATION D'IMAGE VIDEO -

1.1. Définition -

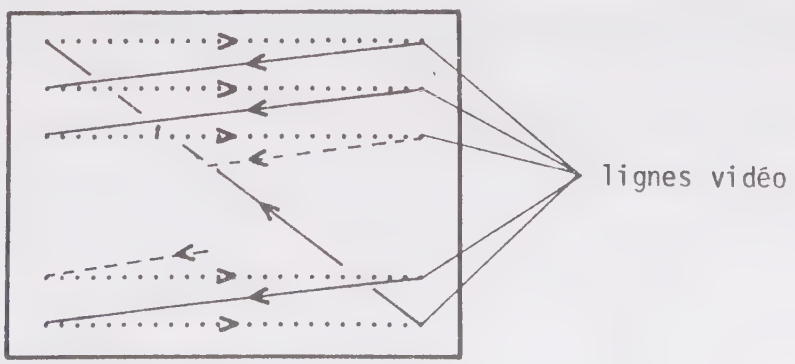
Le processeur d'image à balayage de trame ou "raster scan display" est un système qui permet de générer une image sur un tube cathodique, sous la forme d'un ensemble de points. Chaque point a une luminosité et/ou une couleur spécifiée.

L'image est conservée dans une mémoire d'images sous forme de "Pixel". Chaque pixel représente l'état d'un point (d'une portion) de l'écran : 512 x 512, 1024 x 1024 ou n x m pixels pour représenter tout l'écran.

L'état d'une portion d'écran : allumé-éteint peut au minimum être codé à l'aide d'un bit. Dans le cas d'image monochrome, si l'on veut plusieurs niveaux de gris, l'état d'un pixel sera codé sur p bits (soit 2^p niveaux de gris possible). Dans le cas d'un écran multichrome, on pourra coder ainsi individuellement chaque couleur de base (rouge, vert, bleu).

1.2. Génération de l'image -

Dans un processeur graphique à balayage de trame, le faisceau d'électrons du tube TRC (Tube à Rayon Cathodique) balaie l'écran pour obtenir la trame (succession de ligne vidéo) de l'ordre de 30 fois par seconde, suivant le tracé ci-dessous :



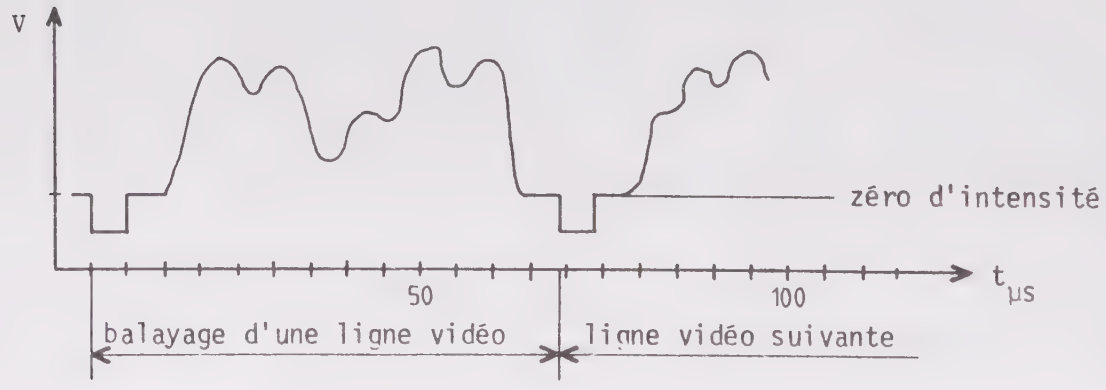
Une ligne vidéo est constituée de points dont on fait varier la luminosité en modulant le faisceau d'électrons. Chaque point de l'écran représente un "pixel" du "raster".

La mémoire d'image est donc balayée périodiquement à la cadence indiquée ci-dessus (30 fois par seconde) de façon à éviter le scintillement de l'image visualisée.

1.3. Techniques de visualisation de l'image -

Pour commander le moniteur TV, il est nécessaire d'avoir un signal dit "signal vidéo" provenant du contrôleur de visualisation.

Le "signal vidéo" est une tension analogique variable dans le temps. A chaque valeur de la tension correspond l'intensité de l'image le long de chaque ligne vidéo.



Modes de balayage -

Il existe deux modes de balayage de la trame :

- mode non entrelacé.
- mode entrelacé.

Mode non entrelacé -

pour générer une image le faisceau d'élection balaye l'écran en une seule fois.

Le faisceau d'électron part en haut à gauche de l'écran et balaye successivement chaque ligne vidéo de gauche à droite. A la fin de chaque ligne vidéo, il retourne au bout à gauche en descendant d'un cran pour commencer à balayer la ligne suivante.

Le signal vidéo module le faisceau d'électrons pendant le balayage de chaque ligne vidéo et c'est en fonction de cette modulation que les points constituant chaque ligne vidéo auront une luminosité variable.

Pendant le passage d'une ligne à une autre, le faisceau est invisible.

Quand le bas de l'écran est atteint, le faisceau retourne à sa position initiale, et le cycle recommence.

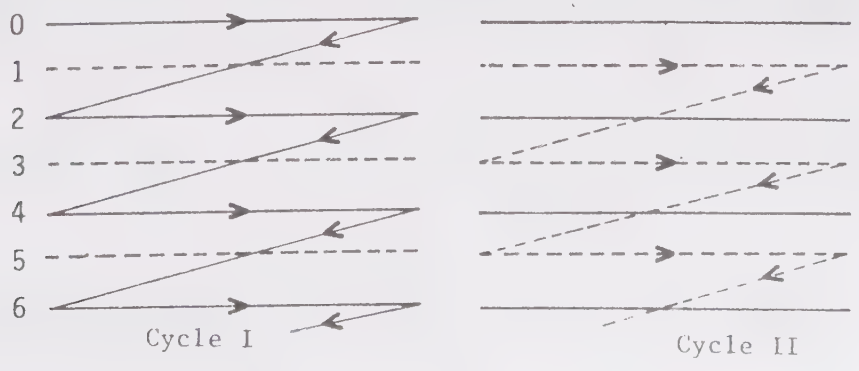
Le nombre de lignes vidéo standard est : 525, 625, 813, 875 et 1023 lignes.

Mode entrelacé -

ce mode est utilisé pour réduire l'effet de scintillement de l'image pendant le balayage de la trame.

Pour générer une image, le faisceau d'électrons balaie l'écran en deux temps : balayage des lignes paires puis balayage des lignes impaires.

Le principe de balayage est le même que pour le mode non entrelacé.



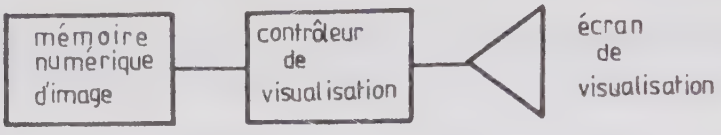
1.4. Architecture de base d'un système graphique à balayage de trame -

Un système graphique à balayage de trame est composé de :

. une mémoire numérique d'image (Image Storage System, Frame Buffer),

. un contrôleur de visualisation (Image Display System, Display Controller)

. d'un écran de visualisation ou "moniteur télévision" (tube à rayon cathodique).



Le contrôleur de visualisation extrait périodiquement l'image de la mémoire, convertit chaque valeur numérique codant un pixel en une tension analogique et génère ainsi le signal vidéo de commande du tube.

Mémoire numérique d'image -

Elle contient toutes les informations nécessaires à la visualisation de l'image à savoir la valeur de l'intensité lumineuse de chaque point ou pixel formant l'image. Sa capacité dépend :

. du nombre de bits par pixel,
. du nombre de pixels par ligne vidéo
. du nombre de lignes vidéo.

Contrôleur de visualisation -

Il lit successivement les données de la mémoire d'image et les convertit sous forme analogique correspondant au signal vidéo, qu'il envoie au moniteur télé.

Le contrôleur de visualisation est généralement conçu pour le mode de balayage entrelacé.

Il génère les différents signaux nécessaires à la visualisation :

. signaux de synchronisation horizontale et verticale (synchronisation composite),

. signal vidéo.

Il assure aussi la gestion de la mémoire d'image :

. raffraîchissement de l'image,

. raffraîchissement des mémoires RAM composant la mémoire numérique d'image,

. écriture de lecture en mémoire.

Ecran de visualisation -

Il "transforme" le signal vidéo en une image.

Le signal de synchronisation ligne commande le générateur de balayage horizontal du téléviseur.

Les signaux de synchronisation trame provoquent le retour du spot en haut de l'écran au début d'une ligne paire ou impaire.

2. DESCRIPTION D'UNE MEMOIRE D'IMAGE -

2.1. Instructions d'écriture et lecture en mémoire -

Avant de visualiser une image, il est nécessaire de charger en mémoire d'image les valeurs de l'intensité lumineuse de chaque point ou pixel.

Le chargement d'un pixel en mémoire ou sa mise à jour se fait par une instruction d'écriture en mémoire, qui range en mémoire la valeur de l'intensité lumineuse de ce pixel repéré par ses coordonnées (x,y).

Il peut être nécessaire de retrouver la valeur d'un pixel. Pour cela, on utilisera une instruction de lecture en mémoire qui renvoie la valeur de l'intensité lumineuse d'un pixel (x,y).

2.2. Mode d'adressage d'une mémoire d'image -

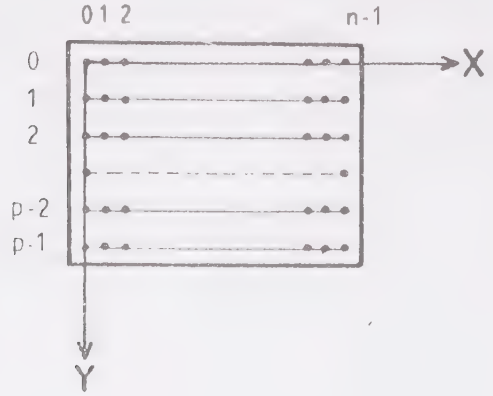
La mémoire logique traduit la position des points ou pixels sur l'écran suivant un système de coordonnées rectangulaire (X,Y). Elle est donc considérée comme une mémoire à deux dimensions. Soient donc :

Y le numéro de la ligne vidéo
X le numéro d'un point situé sur une ligne vidéo.

Convention -

Le point ou pixel de coordonnées (X = 0, Y = 0) se trouve en haut à gauche de l'écran. Soient :

n le nombre de points sur une ligne vidéo
p le nombre de lignes vidéo de l'écran télé



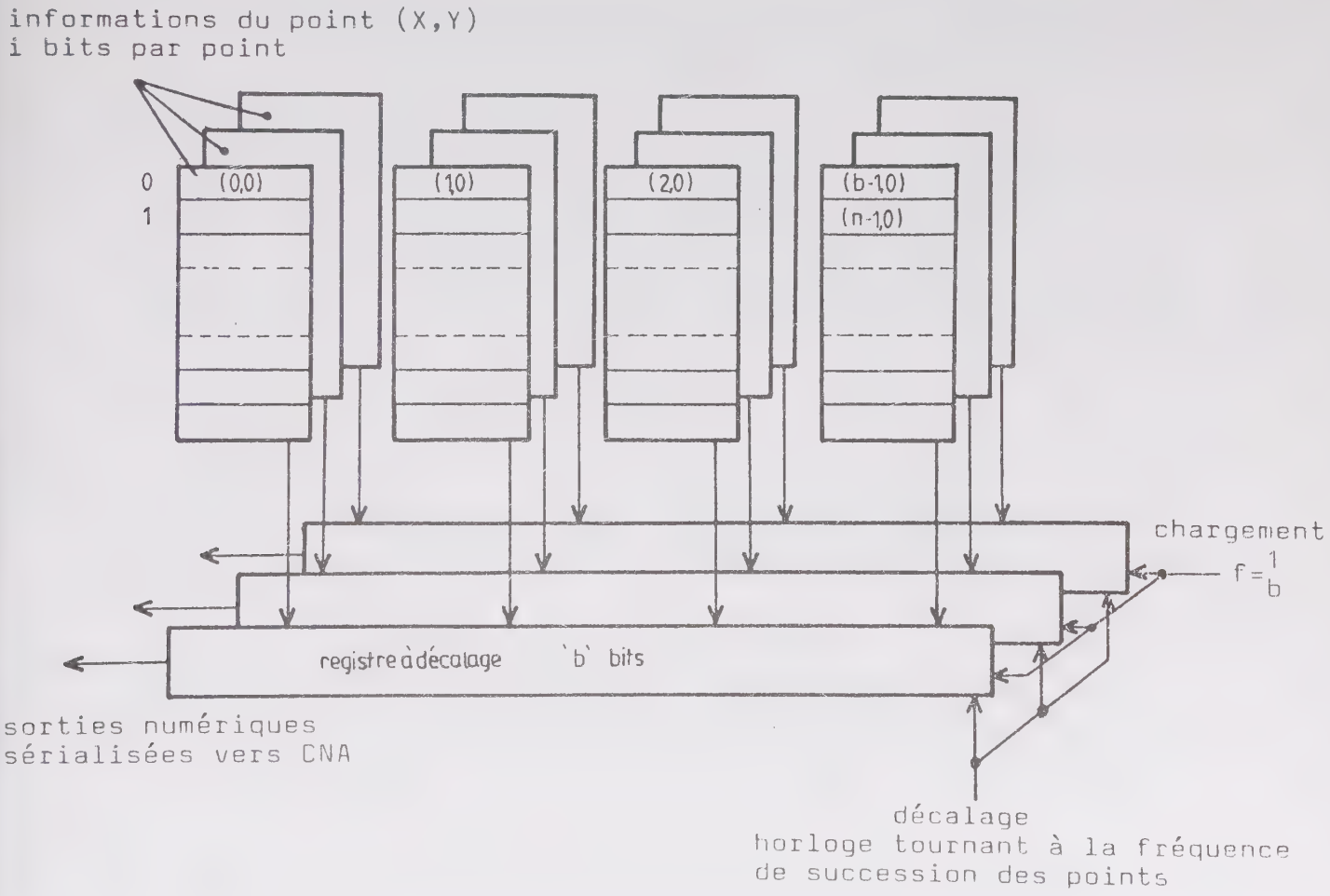
avec X 0, n - 1
Y 0, p - 1

Dans le cas d'une image rafraîchie 25 fois par exemple, une ligne d'un écran de 625 lignes est donc balayée en $\frac{1}{25 \times 625} = 64$.

Une ligne contient environ 640 points, le temps de lecture de deux points voisins est donc $\frac{64}{640} = 100$ ns

Actuellement, le temps de cycle minimum des mémoires est de l'ordre de 270 ns, donc pour le rafraîchissement de l'image, il est nécessaire de lire plusieurs points d'une même ligne vidéo en parallèle et de les sérialiser pour former le signal vidéo.

Une organisation de la mémoire d'image avec le rangement de l'information des points est la suivante :



Physiquement, un point de coordonnée (X,Y) correspond à une adresse mémoire du boîtier et à son numéro.

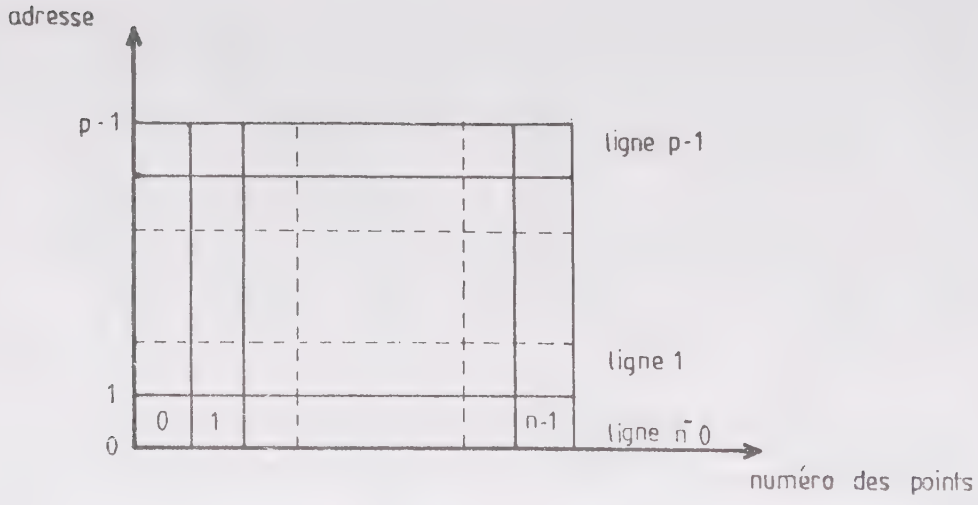
Une solution est de mettre en parallèle autant de boîtiers que de points sur une ligne vidéo :

. les cases mémoires pointées par l'adresse OH contiennent les informations des points sur la ligne vidéo Y = 0

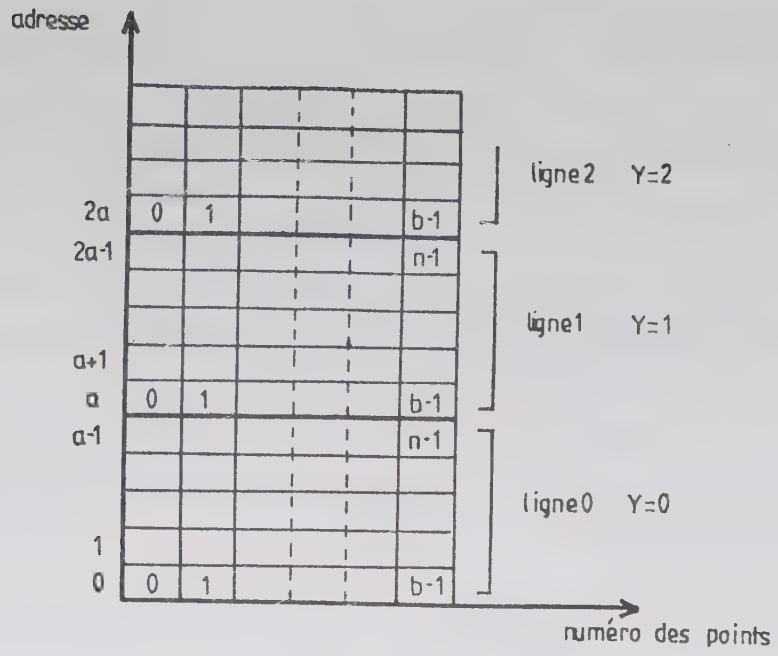
. les cases mémoires pointées par l'adresse 1 H contiennent les informations des points sur la ligne vidéo Y = 1

et ainsi de suite jusqu'à la dernière ligne
à X correspond le rang de boîtier mémoire.

On a donc le graphe de correspondance ci-dessous :



Une autre alternative est de grouper plusieurs points sur une même ligne vidéo, ce qui entraîne un nombre "a" d'adresse mémoire par ligne vidéo.



b étant le nombre de points par ligne d'adresse mémoire.

Calcul de l'adresse mémoire des boîtiers -

Le calcul de l'adresse mémoire est donné par la formule suivante :

$$\text{adresse mémoire} = Y \times \text{nombre d'adresses mémoire par ligne} + \frac{X}{\text{nbre de pts par adresse}}$$

c.à.d. $m = a.Y + \frac{X}{b}$

En effet, soient : n le nombre de points par ligne vidéo
b le nombre de points par adresse mémoire

n est donné par le type d'écran utilisé
b est choisi en fonction du matériel utilisé

On a donc le nombre d'adresses mémoire par ligne vidéo $a = \frac{n}{b}$

Une ligne vidéo contenant "n" points occupe alors "a" adresses mémoires.

Pour une ligne vidéo n° Y donné, pour avoir l'adresse mémoire d'un point numéro X donné, il suffit de diviser X par le nombre "b" de points par adresse mémoire. L'adresse mémoire "m" est la partie entière du résultat de la division de X par b .

Or, quand on incrémente de un le numéro de la ligne vidéo, on ajoute à l'adresse mémoire précédente pour un X donné, le nombre d'adresses mémoire "a" par ligne vidéo.

Une nouvelle ligne vidéo commence toujours par un nombre entier fois le nombre d'adresses mémoires "a" par ligne vidéo.

Suivant le type d'organisation de la mémoire physique, l'écriture ou la lecture de l'état des points dans les cases mémoires pointées par une ligne d'adresses mémoire se fait par l'écriture ou la lecture d'un mot de "b" bits à cette adresse mémoire.

Calcul du rang d'un point sur une ligne d'adresses mémoire boîtiers -

Connaissant le premier point sur une ligne d'adresse mémoire et le numéro d'un point donné sur une ligne vidéo, on peut calculer aisément le rang de ce point.

Pour trouver le rang d'un point sur une ligne vidéo, il suffit de retrancher au numéro de ce point le premier point de l'adresse mémoire "m" considéré.

Pour trouver le premier point de l'adresse mémoire "m" considérée pour $Y > 0$, on fait la différence entre l'adresse mémoire "m" considérée et la première adresse mémoire de la ligne considérée. Le produit de cette différence par le nombre "b" de points par adresse mémoire donne le premier point de cette adresse mémoire.

Le rang "r" d'un point numéro X sur une adresse mémoire est donc :

$$r = X + b \quad a.Y - m \quad = X - b\, E\frac{X}{b}$$

Exemple -

Dans un écran télé en standard américain, on a 640 points par ligne vidéo.

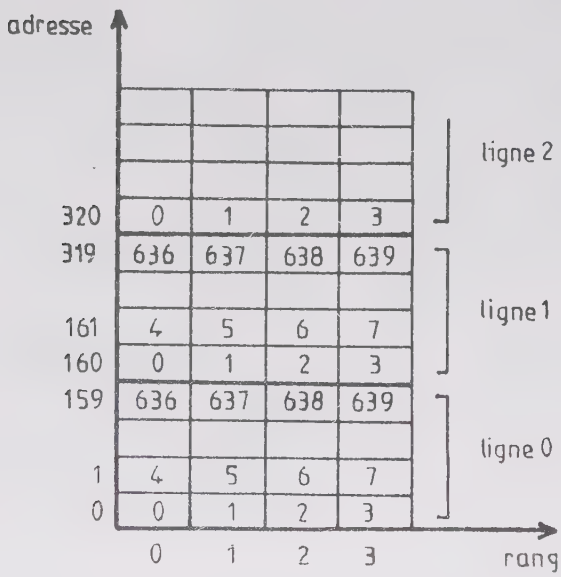
Ce qui donne : n = 640

on prend par exemple

b = 4

ce qui donne $a = \frac{n}{b} = \frac{640}{4} =$ 160 adresses mémoire par ligne vidéo.

Le graphe correspondant est le suivant :



pour (X = 0, Y = 0) on a m = E (0) = 0 r = 0

(X = 638, Y = 0) $m = E\left(\frac{638}{4}\right) = 159$ r = 638 - 4 x 159 = 2

(X = 5, Y = 0) m = 5/4 = 1 r = 5 - 4 x 1 = 1

(X = 6, Y = 1) m = E (1x160+ 6/4) r = 6 + 4 (1 x 60 - 161) = 2

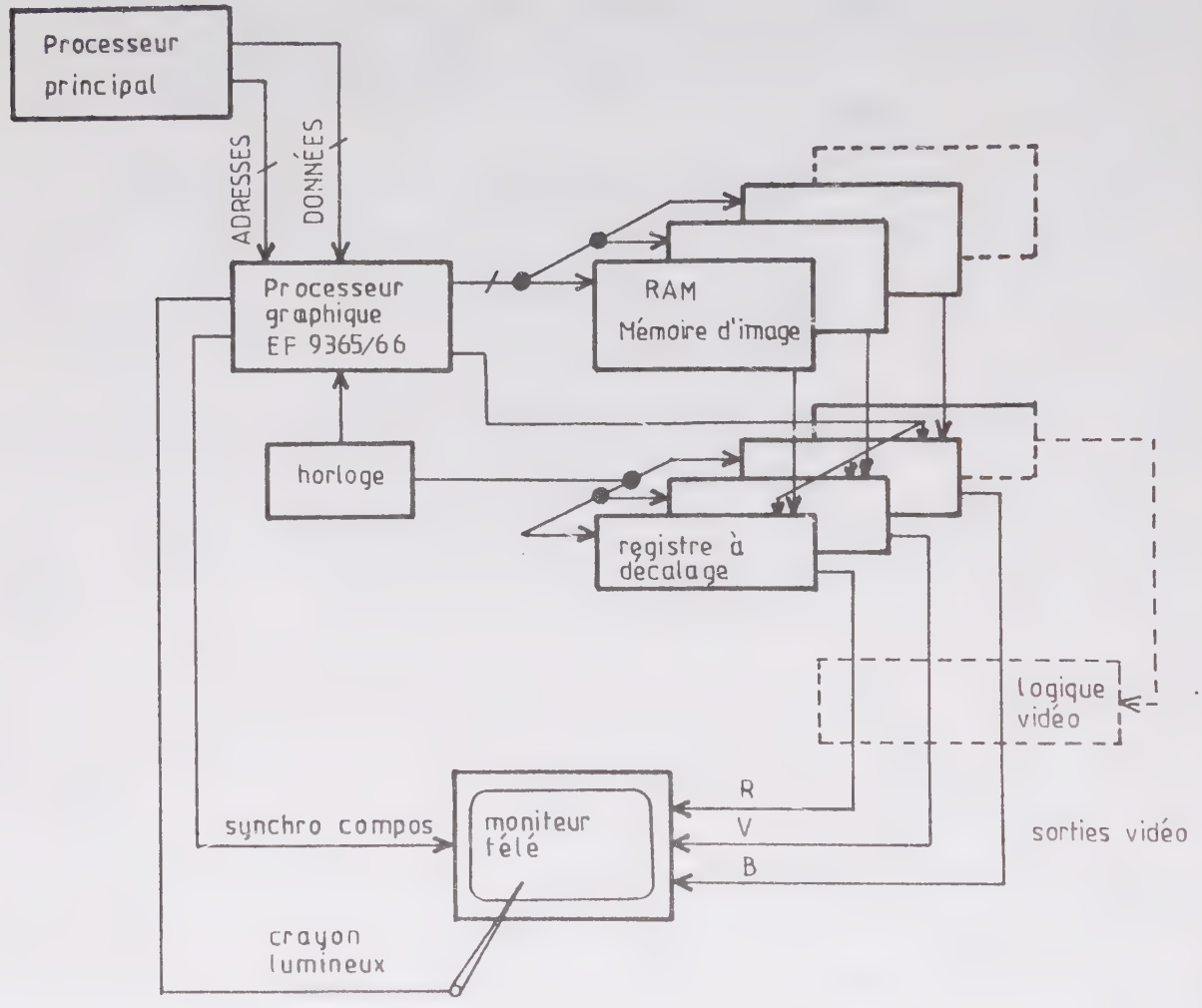
(X = 2, Y = 2) m = 320 r = 2

où E( ) étant la fonction partie entière.

3. DIFFERENTS TYPES DE REALISATION DE SYSTEMES GRAPHIQUES A BALAYAGE DE TRAME -

3.1. Système graphique avec un processeur graphique -

Le système utilise un processeur graphique [Math 78] qui assure la gestion de la mémoire d'image et du moniteur télévision.



Le processeur principal envoie les informations au processeur graphique qui assure :

. la gestion de la mémoire numérique d'image
. la création des signaux de synchronisation du moniteur télévision
. le chargement du registre à décalage.

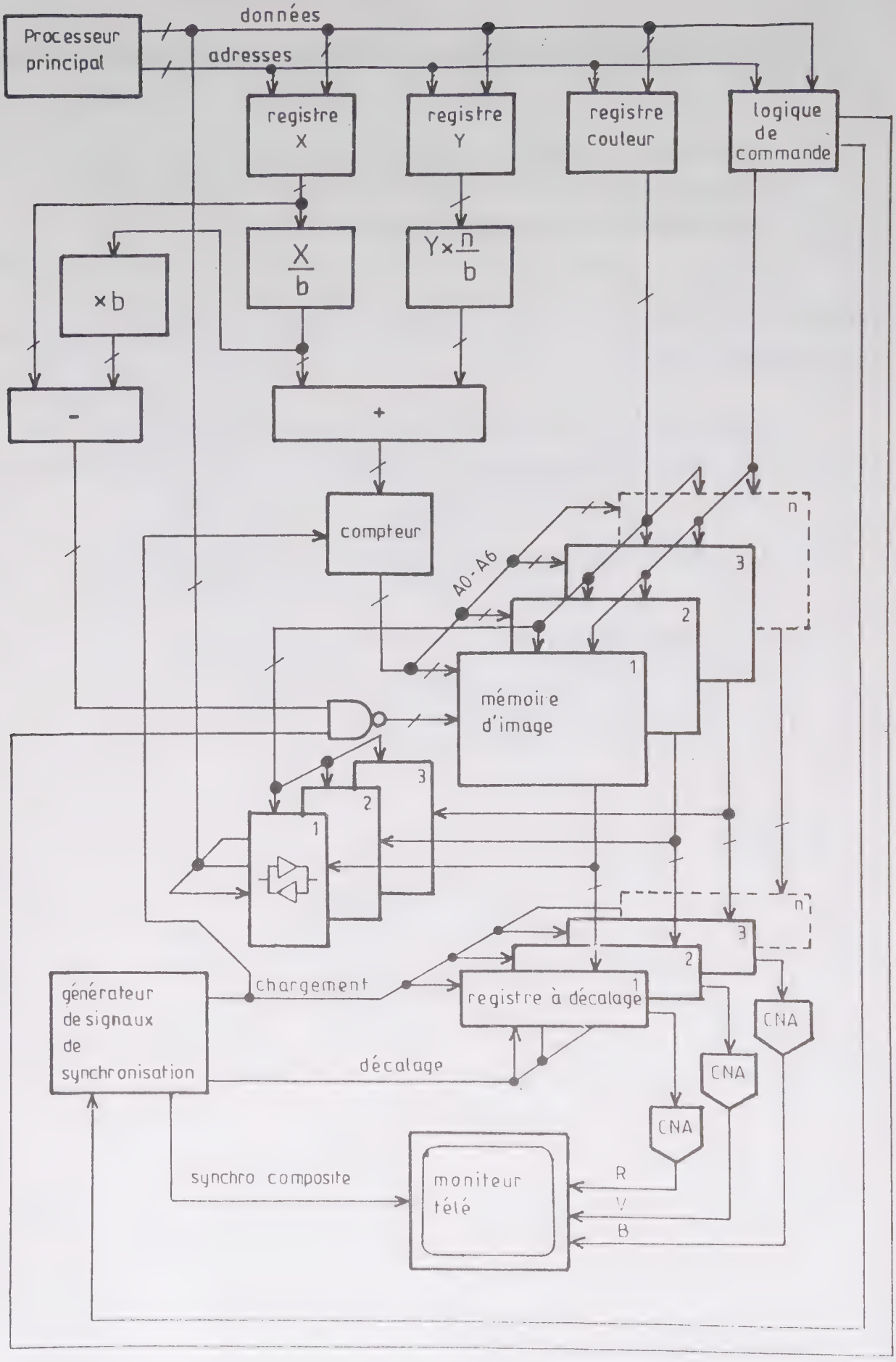
La mémoire d'image est celle qui est décrite dans la partie II. La sélection des boîtiers et le calcul de l'adresse mémoire boîtier sont faits par le processeur graphique.

Les registres à décalage envoient au moniteur télévision, sous forme série les informations des points lus en parallèle lors du rafraîchissement de l'écran.

Le processeur principal est un microprocesseur standard ayant :

. 8 bits de données bidirectionnelles,
. 4 bits d'adresses,
. un signal de lecture/écriture
. une demande d'interruption.

3.2. Système graphique câblé -



Les coordonnées (X, Y), les informations d'un point sont envoyées par le processeur principal sous forme de données et chargées dans les registres X, Y et couleur des points.

On prend l'exemple de 1 bit par point en Noir et Blanc ou en couleurs.

Les opérateurs calculent à partir des données (X, Y) l'adresse mémoire (A_0 - A_7) boîtier, et le boîtier dans lequel se trouve l'information du point (X, Y).

A partir de la donnée "couleur des points", le registre couleur des points choisit la carte mémoire correspondant à la couleur désirée. Par exemple :

- la carte No 1 ou 2 ou 3 pour le noir et blanc
- la carte No 1 pour le bleu
- la carte No 2 pour le vert
- la carte No 3 pour le rouge.

Une fois le calcul de l'adresse mémoire boîtier et du rang boîtier et la sélection de la carte couleur terminés, les informations "niveau d'un point" sont entrées (allumé ou éteint pour 1 bit par point) dans la case mémoire correspondante.

Ceci est valable pour la lecture ou l'écriture d'un point.

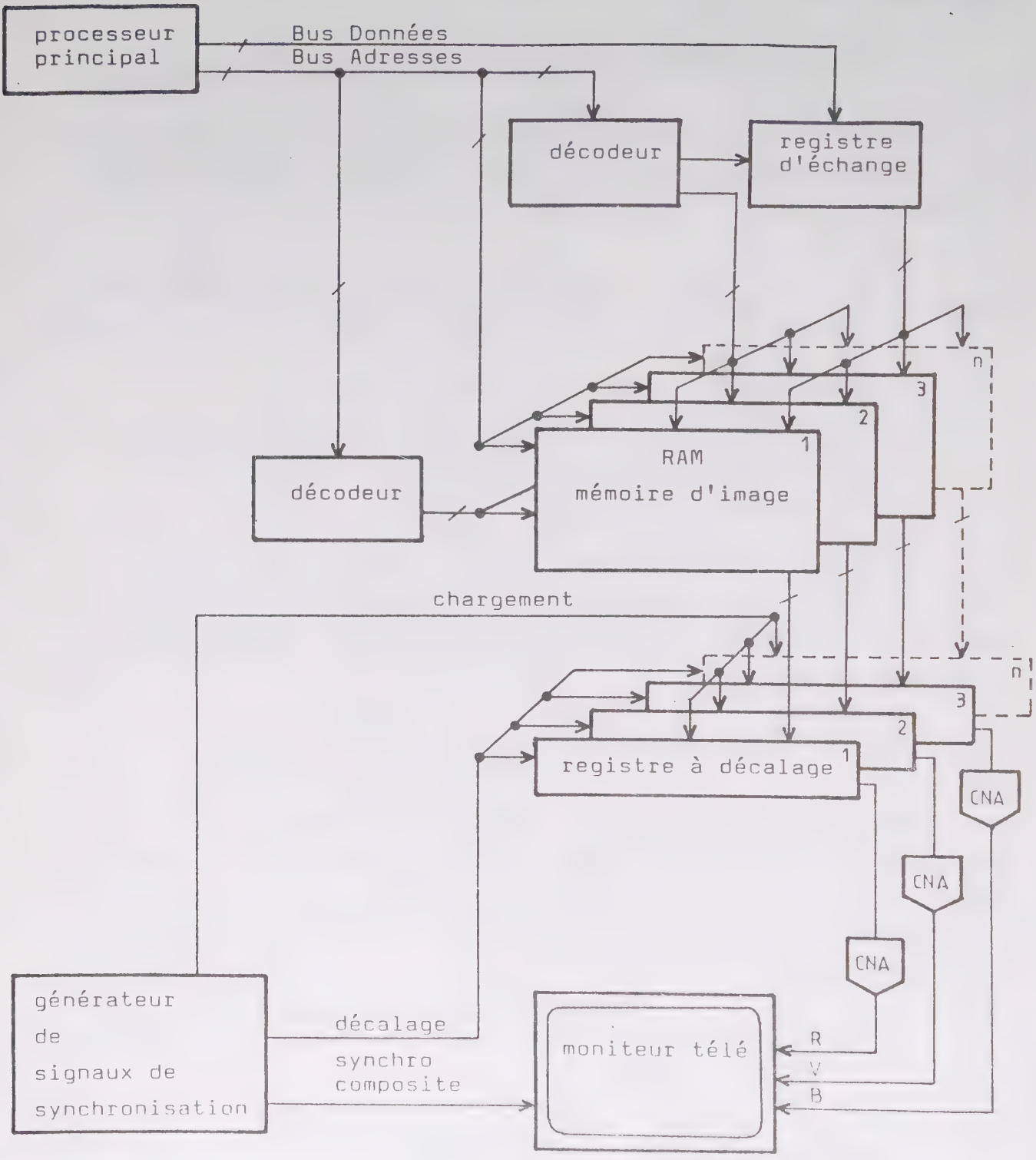
Avant le rafraîchissement de l'image, le compteur est remis à zéro. Ensuite il est incrémenté à la cadence du chargement des registres à décalage (passage d'une ligne d'adresse mémoire boîtier à la suivante). Dans ce mode tous les CS et CE sont à l'état actif c'est-à-dire tous les boîtiers et cartes mémoires fonctionnent.

Le compteur est synchronisé par le générateur de signaux de synchronisation.

Le générateur de signaux de synchronisation assure :

- le chargement et le décalage à la fréquence de succession des points des registre à décalage,
- la génération des signaux de synchronisation du moniteur télévision.

3.3. Système graphique simple -



Comme les informations des points sont rangées séquentiellement dans la mémoire numérique d'image, il est nécessaire de réaliser ces calculs d'adresses et rangement par Software.

Cette structure mobilise un nombre assez important de fils d'adresses.

Les coordonnées (X, Y) ne sont plus entrées sous forme de données, mais sous forme d'adresse.

Les formules établies dans la partie II aident à indiquer dans quelle case mémoire se trouve l'information du point (X, Y).

Le générateur de signaux de synchronisation et le registre d'échange assurent toujours leurs fonctions respectives décrites dans les paragraphes précédents.

CHAPITRE II

SYSTEMES GRAPHIQUES

A MEMOIRE DE TRAMES

SYSTEMES GRAPHIQUES A MEMOIRE DE TRAME

1. INTRODUCTION -

Le présent travail de recherche a consisté en l'étude de l'unité d'affichage

- d'un système graphique à mémoire de trame
- de synthèse d'image
- destiné à la visualisation réaliste d'objets ou de scènes tridimensionnelles.

De façon à bien situer la présente recherche parmi l'ensemble des travaux ayant traits à l'utilisation de terminaux graphiques, interactifs ou non, en CAO, nous en présenterons (ou rappellerons) succinctement ci-après les différentes composantes, à savoir essentiellement :

1) les systèmes graphiques à mémoire de trame,

2) leur utilisation, et en particulier en synthèse d'image,

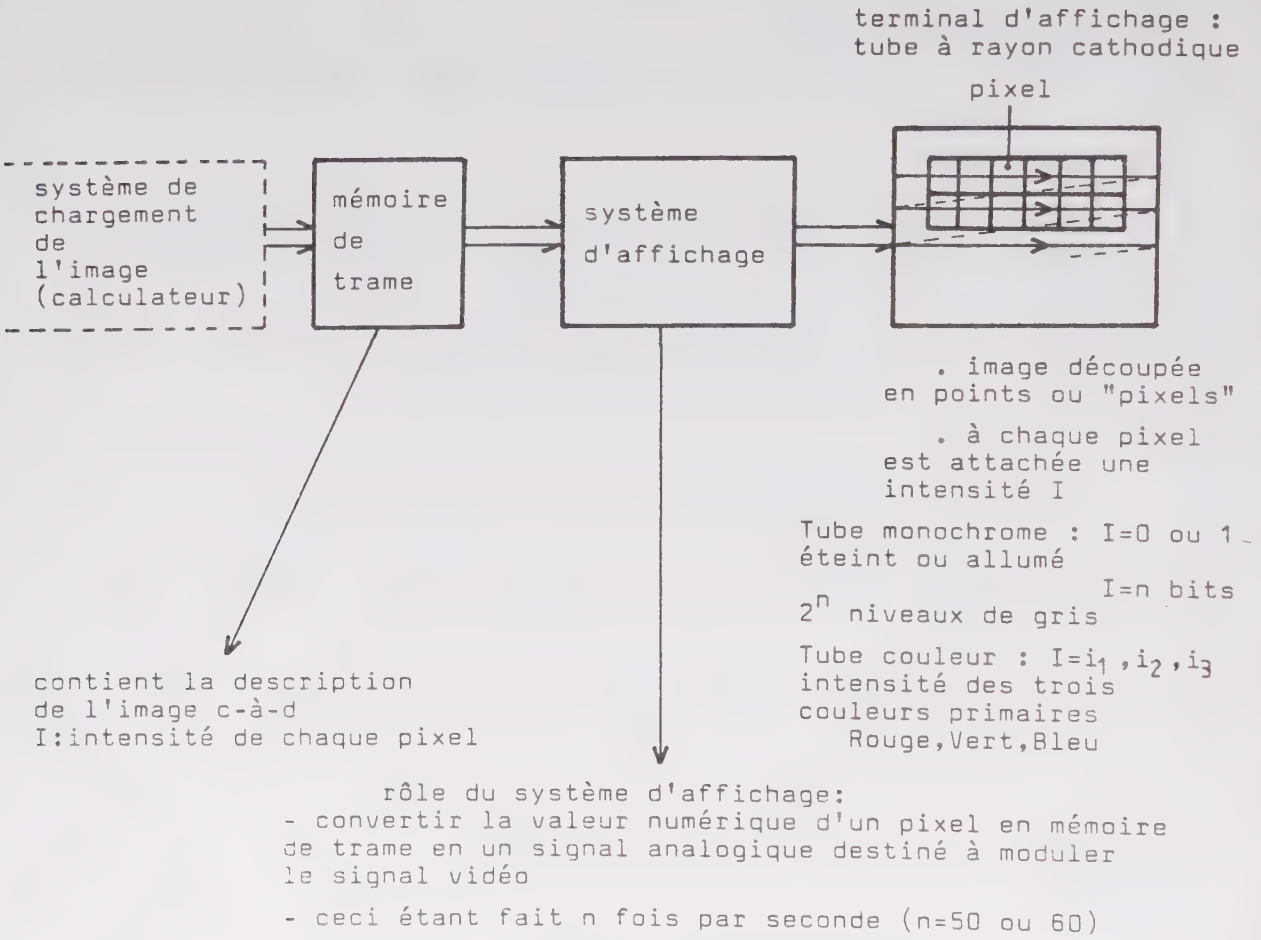
3) les problèmes posés par la visualisation réaliste d'objets tridimensionnels,

4) l'unité d'affichage, objet de la présente recherche.

PRESENTATION SUCCINCTE DES 7 COMPOSANTES

2. SYSTEME GRAPHIQUE A MEMOIRE DE TRAME : Présentation succincte des différents termes utilisés -

Un système graphique à mémoire de trame est composé d'un système de chargement de l'image (calculateur), d'une mémoire de trame, d'un système d'affichage, et d'un terminal d'affichage (tube à rayon cathodique). Voir fig. 1.



Le calculateur contient le modèle informatique de l'image.

La mémoire de trame contient la description de l'image, c'est-à-dire l'intensité I de chaque pixel.

Le système d'affichage a pour rôle de convertir la valeur numérique d'un pixel en mémoire de trame en un signal analogique destiné à moduler le signal vidéo. Ceci étant fait n fois par seconde (n = 30 à 60).

Sur le terminal d'affichage (tube à rayon cathodique), l'image est représentée sous forme de points ou "pixel". A chaque pixel est attachée une intensité I. Pour un tube monochrome I peut prendre les valeurs suivantes :

I = 0 ou 1	un point est allumé ou éteint
I = n bits	un point à 2^n niveaux de gris.

Pour un tube couleur I est une variable à trois dimensions, $I = (i_1, i_2, i_3)$.

(i_1, i_2, i_3) est l'intensité des trois valeurs primaires rouge, vert, bleu.

3. UTILISATIONS DE SYSTEMES A MEMOIRE DE TRAME -

Les principales utilisations de tels systèmes à mémoires de trame sont décrites rapidement ci-après.

3.1. Utilisation en traitement d'image -

Il s'agit dans ce cas :

1) d'acquérir et numériser une image "réelle", par exemple l'aide d'une caméra de télévision,

2) de la visualiser
de la traiter ("manipuler")
de l'analyser.

Exemple de traitement d'image -

- faire apparaître des contours d'objets réels ou fictifs (dessin d'archéologie)
- reconstituer la forme d'objets
- extraire de l'image certaines caractéristiques (ex. degré de maturité d'un lot de pommes).

Pour de telle utilisation, des terminaux graphiques ont été étudiés et offrent au niveau de la console des "opérateurs" spécifiques de manipulation d'image (+, -, log, + , fausse couleur, U, etc.).

3.2. Utilisation en synthèse d'image -

Il s'agit en quelque sorte du problème inverse du précédent : synthétiser une image à partir d'un "modèle", informatique mémorisé dans un calculateur.

Exemples -

. Dessin animé : utilisation similaire à celle de l'artiste qui "peint" c'est-à-dire dépose des "taches" de couleurs sur une toile.

. Simulateur de vol : simulation des conditions d'atterrissage et de décollage.

. Illustrations techniques : affichage de texte de qualité en différentes polices, de traits de différentes épaisseurs, etc.

. Conception assistée par ordinateur.

En Conception Assistée par Ordinateur, l'image est, soit un moyen d'entrée information, soit un moyen de contrôle pour le concepteur ou plus généralement de "communication" entre individu.

Dans le premier cas le concepteur "dessine" sur l'écran, par exemple :

. le schéma d'un circuit électrique ou hydraulique,

. le dessin industriel (vue de face, de dessus...) d'une pièce de mécanique,

. un circuit imprimé. Dans ce cas, le plus souvent "l'objet" est représenté par un dessin fait de lignes et de chaines de caractères (dessin de type "filaire").

Dans le second cas, un grand domaine d'application est la conception d'objets ou scènes tridimensionnelles. Le rôle du système de synthèse d'image

et dans ce cas de générer, à partir d'un modèle décrivant la forme géométrique d'un objet ou d'une scène (ensemble d'objets), une image "réaliste" (que l'objet soit une pièce de mécanique, une chaussure, un immeuble).

Objet de l'étude -

- Dans le premier cas, les consoles actuellement sur le marché résolvent ces problèmes de façon relativement satisfaisante.

- Dans le second cas, de nombreux problèmes existent, tant au niveau des logiciels que des matériels. Il n'existe pratiquement pas de matériels satisfaisants sur le marché sinon à des coûts prohibitifs.

4. VISUALISATION REALISTE D'OBJETS TRIDIMENSIONNELS -

Elle implique :

- un modèle de représentation de l'objet (c'est-à-dire la façon de mémoriser un objet dans un calculateur),

- les procédures de visualisation (c'est-à-dire l'ensemble des techniques permettant de générer une image tridimensionnelle "réaliste").

4.1. Modèle de représentation -

Il existe deux grandes classes de modèles de représentation :

- procédural : dans ce cas l'objet est mémorisé sous la forme d'un "programme".

Exemples : - programme ATP (commande numérique)

- arbre décrivant ces opérateurs volumiques d'un modéleur volumique.

C'est l'exécution de ce programme qui permettra de générer les données caractéristiques de l'objet, par exemple les commandes pour la machine à usiner ou celles pour la machine à dessiner.

- sous formes de données : dans ce cas l'objet est représenté, soit :

a) par décomposition cellulaire (le corps est décomposé en cellules ou volumes élémentaires de formes spécifiques au modèle),

b) par sa frontière : tout objet solide peut être représenté par la frontière (SMA de Smadja), c'est-à-dire ensemble de faces (non forcément planes) limitées par des contours, formées d'arêtes (non forcément rectilignes) limitées par deux points ou sommets.

Dans le cadre de la présente recherche, nous nous sommes appuyés sur cette représentation car :

1. le modèle est général,

2. les travaux d'élimination des parties cachées qui l'utilise sont nombreux,

3. la visualisation réaliste de l'objet nécessite la connaissance de la surface de l'objet (normale en chaque point de la surface).

Partant de la représentation "b", les différentes opérations de visualisation nécessaires pour obtenir une "image réaliste" sont décrites dans le paragraphe suivant.

4.2. Opérations de visualisation nécessaires -

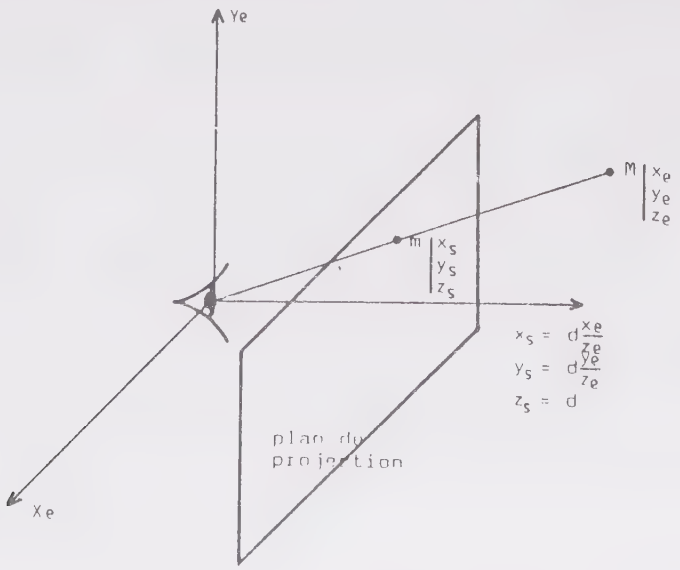
4.2.1. Transformation géométrique (projection, perspective) :

Elle permet de calculer les coordonnées écran (xe, ye) d'un point connu dans l'espace réel (xn, yn, zn). On a donc un passage d'un référentiel de l'univers (xn, yn, zn) au référentiel de l'oeil (xo, yo, zo), ensuite un passage de référentiel de l'oeil au référentiel écran (xe, ye).

Cette transformation nécessite la connaissance :

. de la position de l'oeil
. la normale au plan de projection
. la distance focale d : distance de l'oeil au plan de projection.

Exemple : perspective -

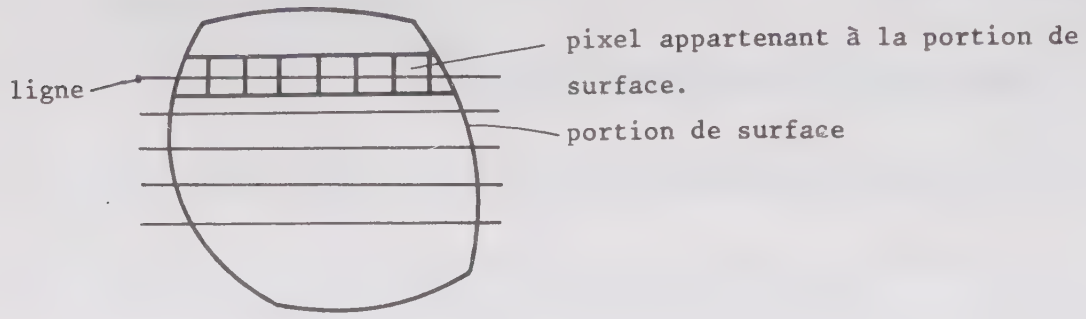


4.2.2. Elimination des parties cachées -

Il s'agit dans ce cas d'éliminer les parties non vues de l'objet (Cf. § 4.3.).

4.2.3. Discrétisation -

Pour afficher une portion de surface sur le terminal, il s'agit de déterminer l'ensemble de pixels (xp, yp) qui devront être visualisés.



4.2.4. Eclairement -

Il s'agit de calculer l'énergie lumineuse émise par tout point P de l'objet. Le modèle d'éclairement utilisé ne doit pas modéliser le monde physique, mais être suffisant pour obtenir une certaine "qualité" de l'image sur l'écran. Il faut distinguer :

- le modèle d'éclairement, défini par l'équation

$E = E_{diffuse} + E_{spéculaire} + E_{transparent}$. E est l'énergie lumineuse perçue par l'observateur. E diffuse est l'énergie lumière émise par chaque point, de manière uniforme suivant toutes les directions.

E spéculaire est l'énergie lumineuse réfléchie par un point éclairé par une ou plusieurs sources ponctuelles quelconques. E transparent est l'énergie arrivant depuis l'intérieur de l'objet. Pour tout point visualisé sur l'écran, on calcule l'intensité lumineuse ip, définie par le modèle optique décrit ci-dessus.

- les algorithmes de calcul de l'intensité I qui impliquent différentes méthodes de "lisage optique".

Remarque sur la visualisation réaliste d'objets tridimensionnels -

Les points les plus pénalisants (qui impliquent le plus de calcul) dans le système de visualisation d'objet tridimensionnel sont :

- l'élimination des parties cachées
- la discrétisation
- les calculs d'éclairement.

4.3. Elimination des parties cachées -

Nous nous limitons au cas des algorithmes spécifiques aux terminaux à mémoire de trame.

4.3.1. Algorithmes dits "de priorité" -

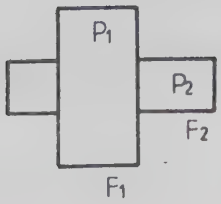
Ils utilisent le fait que si une face est visualisée sur un écran (c'est-à-dire discrétisée et mémorisée en mémoire de trame), la visualisation de la face suivante "écrasera" éventuellement la face déjà mémorisée. Ces algorithmes essaient donc de déterminer un ordre total entre les faces. La profondeur Zi attache à chaque face une priorité Pi.

Si une face Fi a une priorité supérieure à la face Fj alors on a la relation suivante : "Fi n'est pas cachée par Fj."

Exemple : algorithme de Newell -

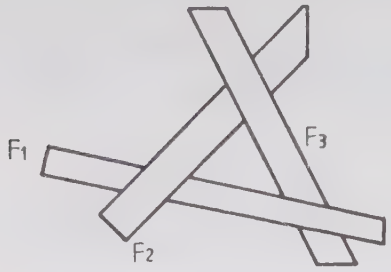
Il s'applique à des faces polygonales planes. A la fin de l'algorithme on a un ensemble de polygones tels que :

Fi de priorité "supérieure ou égale" à Fj
alors Fi n'est pas caché par Fj (tous les pixels de Fi auront la même priorité Pi).
Ceci est illustré par la figure suivante :

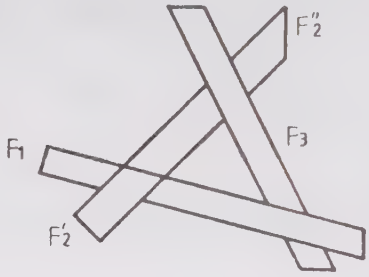


Fi a une priorité supérieure à F2 alors F1 n'est pas caché par F2.

Il existe de nombreux problèmes et cas particuliers tels que la figure suivante l'illustre.



Il est donc nécessaire de découper la face F2 ou F3 en deux sous-faces. Choisissons par exemple la face F2 qui va être découpée en deux sous-faces F'2 et F"2.



Enoncé de l'algorithme de visualisation -

Pour chaque face de priorité P, calcul de tous les pixels j : xj, yj et intensité Ij, P étant constant pendant le calcul (gous les pixels ont la même priorité).

4.3.2. Algorithme par ligne de balayage -

Le principe est à peu près le même que l'algorithme de priorité. Mais au lieu de s'appliquer sur des faces planes, il s'applique sur des segments obtenus, par projection de l'intersection d'un plan de balayage avec les faces, sur la ligne de balayage correspondant. On appelle ligne de balayage la projection sur la droite intersection du plan de l'écran et du plan de balayage.

Exemple : algorithme de Watkins

ligne de balayage

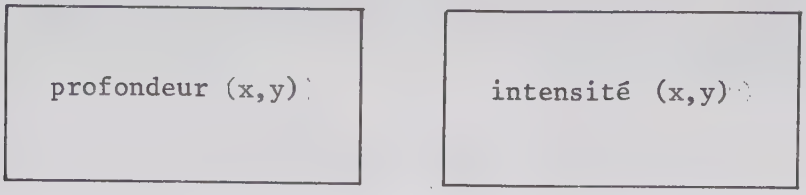
plan de l'écran

plan de
balayage

La relation est la même : L_1 de priorité P1 n'est pas caché par L2 de priorité

4.3.3. Algorithme de Z Buffer -

Utilisation d'un buffer Z dans lequel on conserve pour tout pixel de l'image visualisée sur l'écran, la profondeur Z. L'algorithme dans ce cas est relativement simple.



Z Buffer mémoire de trame

Enoncé de l'algorithme -

* phase d'initialisation

- pour tout pixel (x,y) faire

profondeur (x,y) ← 1
intensité (x, y) ← intensité de fond de l'image

* pour toute face de la scène faire

début
pour tout pixel (x,y) faire
début
calculer Z (x,y)
si Z profondeur (x,y) alors
début profondeur (x,y) Z
calculer I(x,y)
intensité (x,y)
fin
fin
fin

Remarque -

Cet algorithme est valable

- quel que soit le type de la face (plane ou non)
- quel que soit le type d'objet (surfacique, volumique, etc.)

4.3.4. Comparaison des deux méthodes -

Dans le cas d'objets de même nature, approximés par des faces planes :

	Newell	Z Buffer
Tri des faces	X	
Discrétisation (x,y)	X	X
Calcul de Z		X par ex. faire le calcul de Z pour les 1024 x 1024 pixels
Calcul d'intensité	X	X

5. UNITE D'AFFICHAGE -

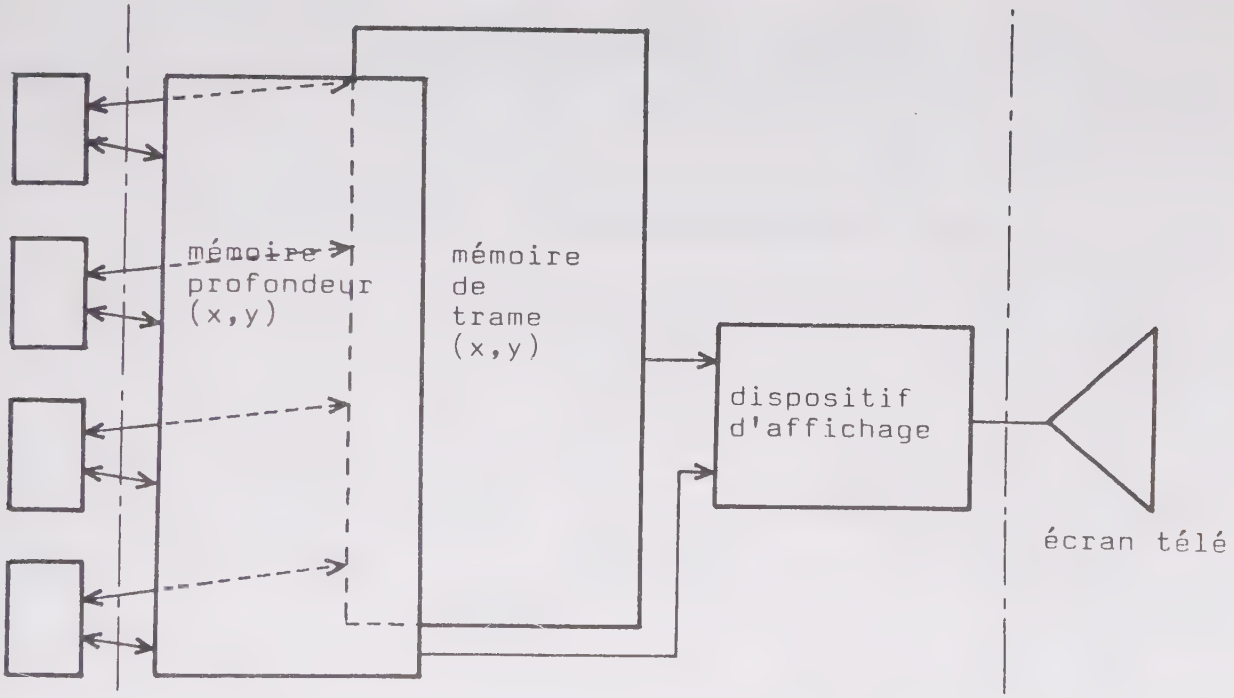
5.1. Accélération des algorithmes -

5.1.1. Parallélisation des calculs -

Dans les deux méthodes (Newell et Z Buffer), les phases les plus évidentes à paralléliser sont :

- la discrétisation
- le calcul de Z
- le calcul d'intensité

Ces phases peuvent être faites à condition de disposer d'un "buffer Z". D'où la structure suivante :



Unité d'affichage

5.1.2. Problèmes -

Les problèmes les plus évidents sont :

- l'accès en lecture et écriture simultanée à la mémoire profondeur Z, par les "n calculateurs" (spécialisés ou non) ;

- l'accès en écriture à la mémoire de trame par les "n calculateurs" ;

- l'accès en lecture à la mémoire de trame par le dispositif d'affichage qui occupe la plus grande partie du temps d'accès à la mémoire de trame.

5.2. Principe de l'unité d'affichage -

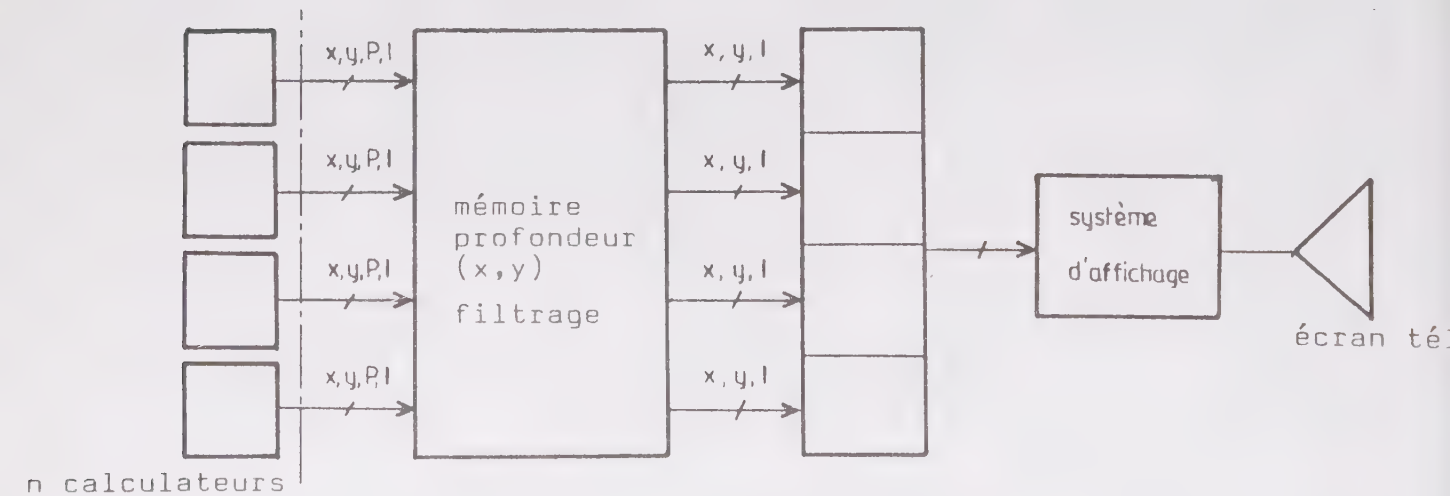
5.2.1. But -

Le but de l'étude est :

- ne pas ralentir les "n calculateurs", qui disposeront d'un accès simultané à la mémoire de profondeur Z et d'intensité I par bus séparé) ;

- avoir la plus grande bande passante pour le chargement de la mémoire d'intensité I (a priori, comme on le verra plus loin, les temps de chargement de la mémoire sont très faibles).

Ce but nous amène à la structure de principe suivant :



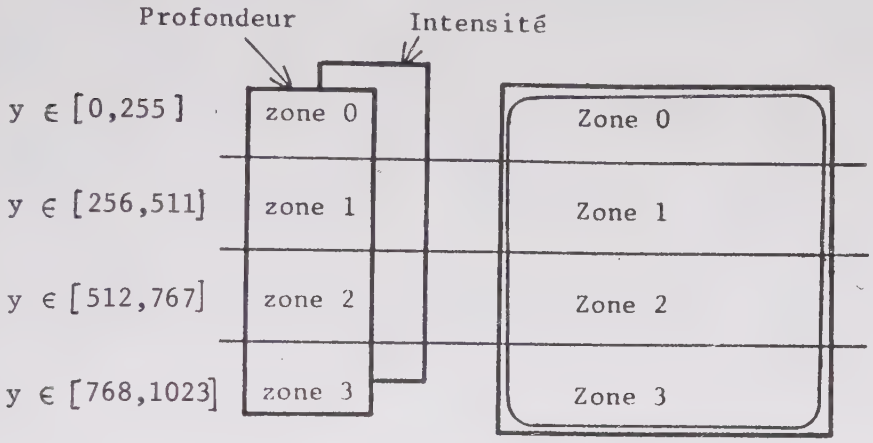
Unité d'affichage

- la mémoire de trame est découpée en zones
- chaque calculateur envoie par bus séparé à la mémoire Z les informations pixels (x,y,P,I). La mémoire Z contient la profondeur de

chaque pixel (x,y) et filtre chaque information provenant des calculateurs en l'envoyant dans la zone mémoire I correspondante, après avoir fait la comparaison des profondeurs.

5.2.2. Organisation de la mémoire -

Les deux mémoires profondeur et intensité sont découpées en zones indépendantes suivant l'ordonnée écran"y"(y correspond au numéro d'une ligne). Ceci est illustré par la figure suivante avec par exemple y appartenant à [0,1023] et un nombre de zones égal à 4.



écran

CHAPITRE III

ARCHITECTURE DE L'UNITE D'AFFICHAGE

ARCHITECTURE DE L'UNITE D'AFFICHAGE

1. PRESENTATION ET CARACTERISTIQUES DE L'UNITE D'AFFICHAGE -

L'unité d'affichage permet de ne pas ralentir les "n calculateurs" et d'avoir une plus grande passante pour le changement de la mémoire de trame. La mémoire de trame est divisée en plusieurs zones indépendantes accessibles par tout calculateur.

Les caractéristiques de l'unité d'affichage sont :

- possibilité de définir le nombre de zones de 1 au nombre total de lignes 512 ou 1024 (par puissance de 2),
- deux résolutions possibles :
  - . 1024 x 1024 non-entrelacé (noir et blanc)
    entrelacé (noir et blanc, couleur)
  - . 512 x 512 non-entrelacé (noir et blanc, couleur)
    entrelacé (noir et blanc)
- les données transmises sont (X, Y, P, I) où : (X,Y) sont les coordonnées écran d'un point courant, P est la profondeur de ce point, I est l'intensité lumineuse de ce point (I a été défini au § 2 du chap. II). Le nombre de bits par couleur est défini à la construction de l'appareil.
- deux modes de travail possibles : mode image ou mode espace.

1.1. Mode de travail image -

L'unité d'affichage offre une modularité par "image". L'unité d'affichage permet de visualiser une image parmi 64. Il est possible de changer une image pendant qu'une autre est visualisée sur l'écran.

Deux commandes sont disponibles pour le numéro de l'image à visualiser d'une part et le numéro de l'image à charger d'autre part.

Les données transmises sont (X,Y,P,I) avec (X,Y) appartenant à $[0,1023]^2$ ou $[0,512]^2$, P appartenant $[0,n-2]$ et I appartenant à $[0,K-1]$.

1.2. Mode travail espace -

L'écran est une fenêtre se déplaçant de façon discrète dans un espace d'adressage de 8192 x 8192. Le pas est égal à [0,511] ou [0,1023] .

Il est possible de charger une fenêtre tandis qu'une autre est en train d'être visualisée.

La commande origine permet de spécifier l'origine (Xo,Yo) de l'espace à visualiser.

Les données sont toujours (X,Y,P,I) avec seulement (X,Y) appartenant à $[0,8191]^2$, le reste étant inchangé.

2. ORGANISATION DE L'UNITE D'AFFICHAGE -

L'unité d'affichage comporte deux organes distincts :

- l'organe de mise à jour de l'image
- l'organe vidéo.

L'organe de mise à jour de l'image est composé :

- de n tampons récepteurs recevant les données (X,Y,P,I) provenant d'un calculateur correspondant,

- de n circuits de décodage. Chaque circuit de décodage détermine dans quelle zone va être rangée l'information (X,Y,P,I) : entrée du circuit décodage (X,Y,P,I), sortie (X,Y,P,I,z), z étant la zone où (X,Y,P,I) va être rangée,

- d'un réseau d'interconnexion. Il permet de relier une quelconque des n entrées à une quelconque dez z sorties. L'information (X,Y,P,I,z) se trouvant à une entrée n_i, se trouve à la sortie z,

- de z tampons émetteurs. Chaque tampon émetteur reçoit pour une zone donnée les informations (X,Y,P,I)z correspondantes,

- de z circuits de mise à jour de l'image associés à z registres ancien point. Chaque circuit de mise à jour s'occupe

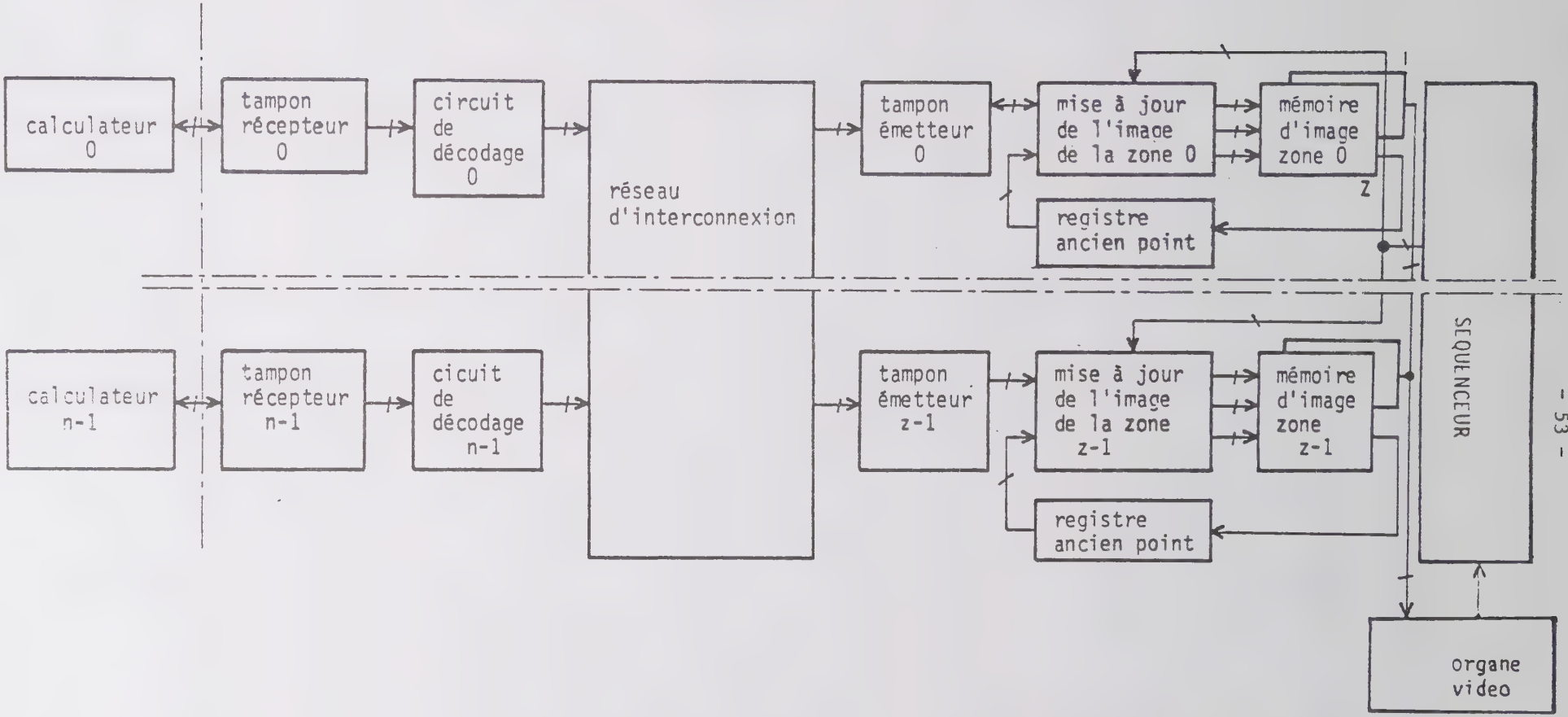
. de la lecture-écriture de la profondeur P et de l'écriture de l'intensité I d'un point de coordonnées écran (X,Y) dans la mémoire de profondeur et dans la mémoire d'intensité respectivement. Seul un point est adressé,

. du rafraîchissement des mémoires d'images de la zone,

. de la visualisation d'une image "b" points (dans le cas de 1024 x 1024 "b" = 16) de la mémoire d'intensité sont adressés simultanément et envoyés vers l'organe vidéo sur un bus unique,

- d'un séquenceur qui envoie en permanence le numéro de zone en train d'être visualisée.

L'organe vidéo sérialise ces "b" points lus simultanément pour former le signal vidéo.



ORGANISATION DE L'ORGANE DE MISE A JOUR DE L'IMAGE

3. ORGANE DE MISE A JOUR DE L'IMAGE -

3.1. Principe de fonctionnement -

La mémoire d'image est divisée en zones. Dans chaque zone on a $\frac{Dv}{z}$ lignes si Dv est la définition verticale (256, 480, 512, 1024). Les lignes sont rangées par ordre croissant dans chaque zone. Par exemple :

la zone 0 contient les lignes 0 à $\frac{Dv}{z} - 1$

" 1 " " $\frac{Dv}{z}$ à $\frac{2Dv}{z} - 1$

" z-1 " " $(z-1)\frac{Dv}{z}$ à $Dv - 1$

Les calculateurs envoient les informations au circuit d'aiguillage sans attendre qu'une information soit aiguillée ou non. Ceci nécessite un tampon de réception pour chaque calculateur, au niveau du circuit d'aiguillage. Quand un tampon est plein, il envoie un signal au calculateur correspondant pour l'arrêter. Dès que le tampon est prêt à recevoir une nouvelle information le calculateur correspondant redémarre.

Les informations contenues dans les tampons sont décodées en parallèles suivant leurs ordonnées y. Ensuite chaque information est mise dans le tampon d'émission de la zone correspondante car une information peut arriver pendant qu'une mise à jour de l'image est en train de se faire. Les tampons récepteurs et émetteurs jouent le rôle de file d'attente.

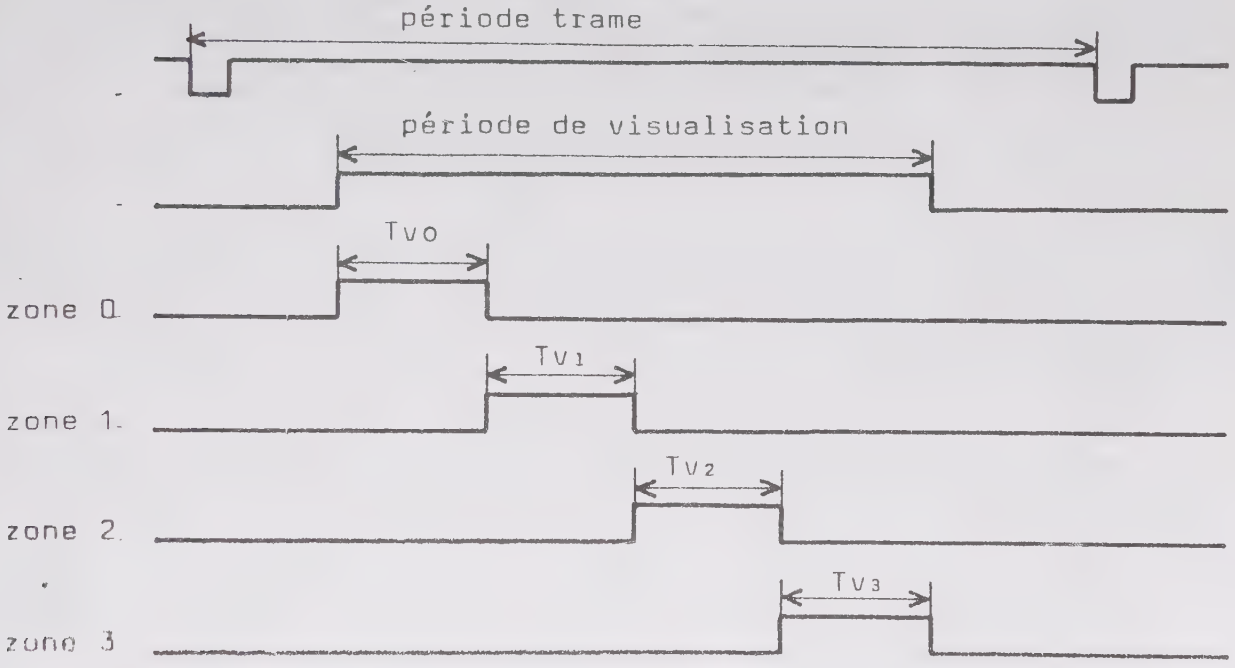
La mise à jour d'une zone est indépendante de la mise à jour des autres zones. Pour chaque zone, on distingue trois périodes distinctes :

- période de mise à jour de l'image
- période de visualisation
- période de rafraîchissement des mémoires.

Si Tv est la période de visualisation globale et Tvi la période de visualisation par zone, on a alors $Tvi = \frac{Tv}{z}$. Donc plus le nombre de zones est grand, plus sa période de visualisation est petite, plus la période d'écriture en mémoire est grande (voir figure illustant le cas d'une définition de 512 x 512 non-entrelacé).

Le séquenceur envoie en permanence aux circuits de mise à jour de l'image le numéro de la zone mémoire qui est en train d'être visualisé. Pendant ce temps, il ne se passe rien au niveau de la zone correspondante, c'est-à-dire que son circuit de mise à jour de l'image arrête toutes les opérations en cours pendant que le tampon émetteur se remplit.

Pendant la période de visualisation Tv, le séquenceur envoie aux circuits de rafraîchissement de l'image les données correspondant à T_{v1} puis T_{v2} et jusqu'à Tv_z si z est le nombre de zones et on repasse à Tv1 et ainsi de suite.

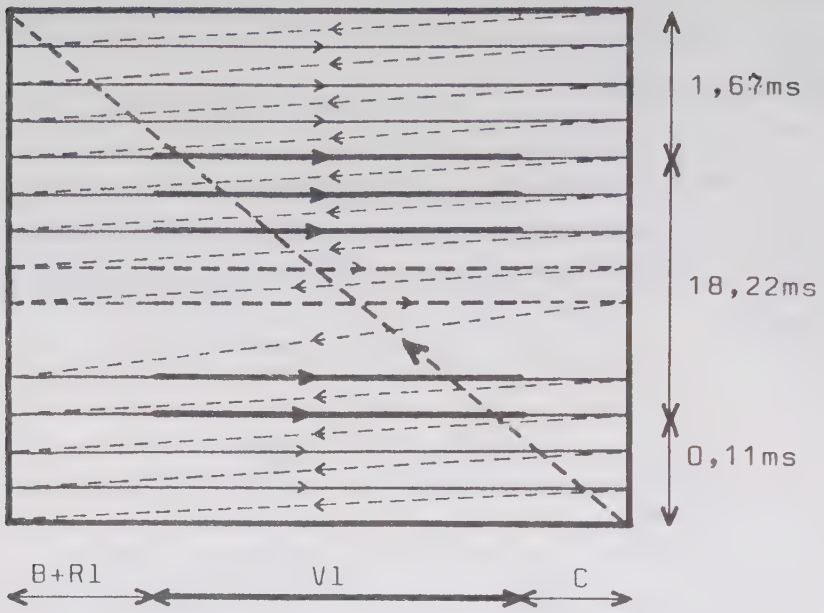


Répartition des périodes de visualisation pour
4 zones mémoires

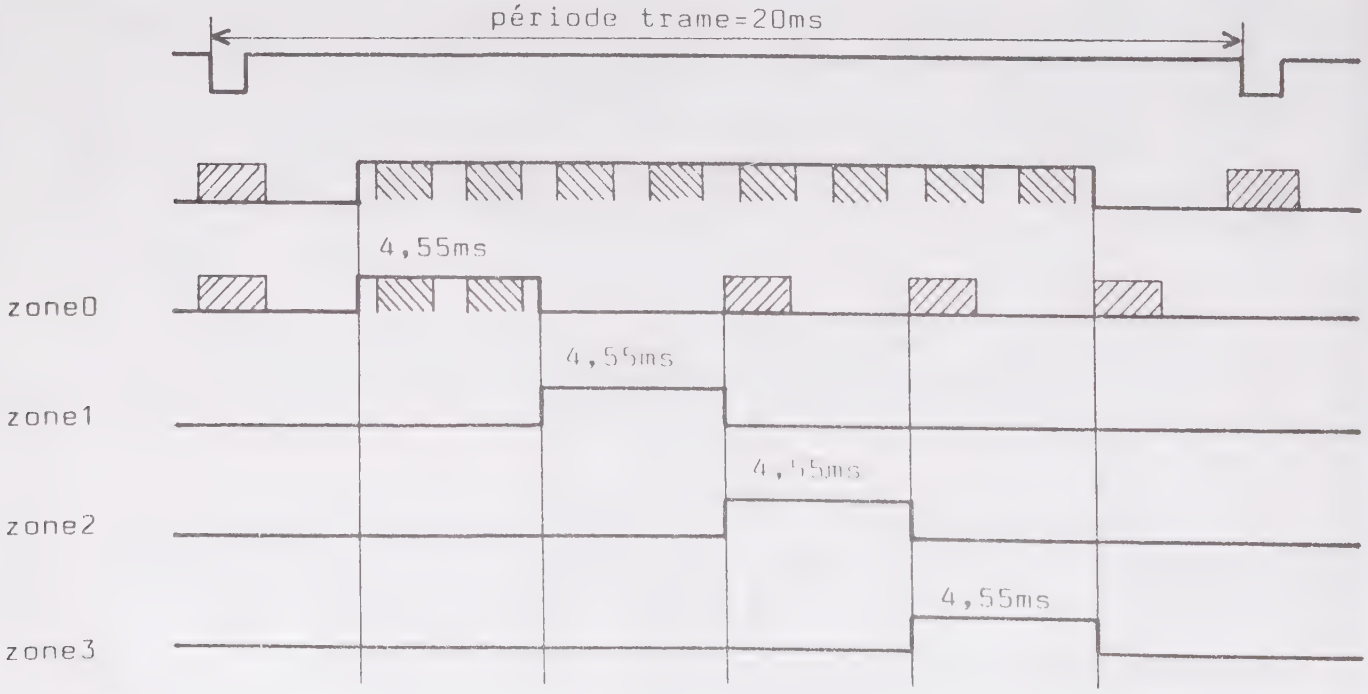
Le principe de la mise à jour de l'image est la suivante : on suppose qu'on est dans la période d'autorisation de mise à jour de la zone et on suppose aussi que l'on travaille dans une zone donnée. Le principe est le même pour les autres zones.

Exemple:-512x512 non entrelacé

V_1=22,8 µs
$B+R_1$=10,7 µs
C=2,1 µs
Tr≃91,2 µs



V:visualisation et rafraichissement implicite≃11,7ms
R:rafraichissement≃91,2µs
E:écriture≃8,25ms



V≃2,9ms
R≃0,4ms
E≃16,7ms par zone

Le circuit de mise à jour de l'image fait l'acquisition d'un nouveau point, transforme ces coordonnées X, Y sous forme d'adresse mémoire (le principe de transformation est donné dans la partie I) et l'ancien point correspondant est mis dans le registre ancien point par lecture de la mémoire à cette adresse. Ensuite les bits de profondeur sont comparés, et celui qui a la profondeur la plus faible est écrit dans la zone mémoire d'image et on passe au point suivant et le processus recommence. En cas d'égalité de la profondeur, le dernier arrivé est chargé dans la zone mémoire d'image.

En résumé :

. acquisition du nouveau point
. lecture et chargement de l'ancien point dans le registre ancien point
. Comparaison des profondeurs
. Ecriture en mémoire suivant la profondeur
. Passage au point suivant.

3.2. Organisation de la mémoire d'image -

La mémoire d'image est organisée de façon à lire plusieurs points en parallèle.

Lors de l'utilisation du système, il est parfois commode pour l'utilisateur, soit

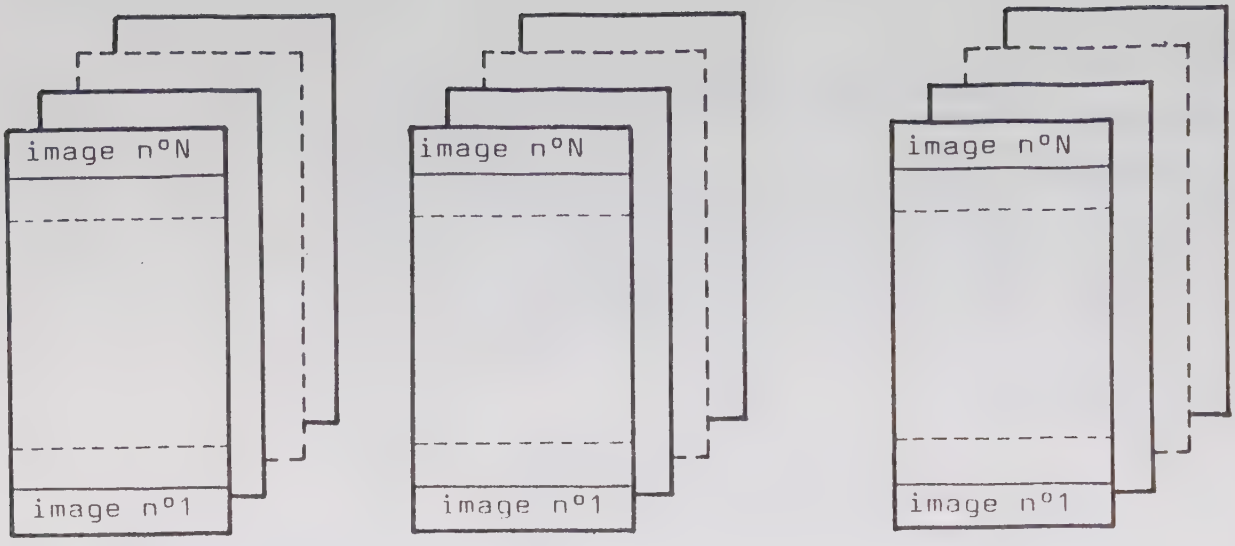
- d'avoir un espace adressable dans lequel on a une fenêtre de visualisation mobile,

- de retrouver les images originales, les images sur lesquelles on a fait une modification et les images finales.

Pour ces besoins et pour augmenter la rapidité d'écriture au niveau mémoire d'image on a choisi l'organisation suivante :

- dans le premier cas, l'organisation de la zone est la même que celle donnée dans la partie Principes généraux d'un système graphique à balayage de trame,

- dans le second cas, on a aussi la même organisation mais dans chaque zone on a plusieurs images.



Organisation d'une zone mémoire dans le cas où on a plusieurs images.

Note : pour certaines mémoires, le nombre d'images est égal au nombre de zones.

Justification des choix des mémoires 64 K :

- vitesse assez rapide mode paye
- faible consommation de courant ?
- avoir une grande capacité mémoire tout en minimisant la densité des composants.

Compte tenu de l'évolution technologique sur les circuits intégrés à grande échelle et l'apparition des mémoires 64 K dynamiques, le choix des composants se porte sur cette dernière permettant ainsi d'avoir une grande capacité mémoire avec une densité de composants réduite.

3.2.1. Rangement des points dans la mémoire d'image -

Mode de travail espace -

Dans le mode de travail où on a un espace adressable, le rangement des points dans la mémoire est fonction du nombre de zones. L'espace adressable

choisi étant 8192 x 8192 et le choix des mémoires étant porté sur les mémoires 64 K RAM dynamiques qui ont 256 rangées et 256 colonnes.

Le rangement étant donné par les tableaux suivants en prenant le nombre de zones z comme paramètre.

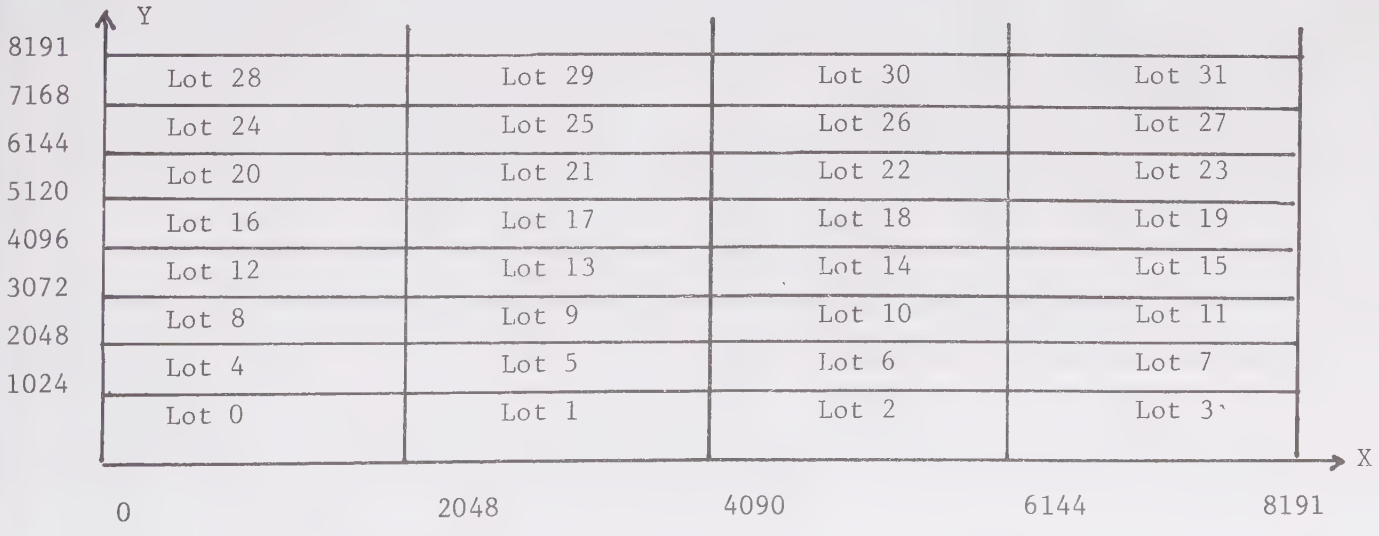
Nota : un bloc correspond à 16 boîtiers mémoires lus en même temps pour le rafraîchissement de l'image.

Un lot est un ensemble de n blocs.

2 zones : il est composé de 32 lots de 2 zones. L'affectation des points dans les blocs est la suivante (en ordonnées on a les colonnes).

C	lot 0		lot 1	
255	y = 0 à 511 x = 1024 à 2047	y = 512 à 1023 x = 1024 à 2047	y = 0 à 511 x = 3072 à 4095	y = 512 à 1023 x = 3072 à 4095
128 127	y = 0 à 511 x = 0 à 1023	y = 512 à 1023 x = 0 à 1023	y = 0 à 511 x = 2048 à 3071	y = 512 à 1023 x = 2048 à 3071
0	zone 0	zone 1	zone 0	zone 1

et ainsi de suite jusqu'au 32 ème lot. L'affectation des lots étant donnée par le graphe suivant en fonction de X et Y.



Au niveau de la mémoire (boîtier), pour un lot et une zone donnés on a le tableau suivant, par exemple lot 0, zone 0, qui donne R et C (adresses rangée et colonne) en fonction de X et Y.

Rangées d'un bloc de 16 mémoires / colonnes d'un bloc de 16 mémoires	0	1	63	64	255
0	x = 0 à 15 y = 0	x = 16 à 31 y = 0	1008 à 1023 y = 0	0 à 15 y = 1	1008 à 1023 y = 3
1	x = 0 à 15 y = 4	x = 16 à 31 y = 4	1008 à 1023 y = 4	0 à 15 y = 5	1008 à 1023 y = 7
127	x = 0 à 15 y = 508	x = 16 à 31 y = 508	1008 à 1023 y = 508	0 à 15 y = 509	1008 à 1023 y = 511
128	x = 1024 à 1039 y = 0	x = 1040 à 1053 y = 0	2032 à 2047 y = 0	1024 à 1039 y = 1	2032 à 2047 y = 3
129	x = 1024 à 1039 y = 4	x = 1040 à 1053 y = 4	2032 à 2047 y = 4	1024 à 1039 y = 5	2032 à 2047 y = 7
255	x = 1024 à 1039 y = 508	x = 2030 à 2047 y = 508	2032 à 2047 y = 508	1024 à 1039 y = 509	2032 à 2047 y = 511

d'après les tableaux ci-dessus, les formules donnant L, R, C sont donc :

$$L(2) = 4\,E\left\{\frac{Y}{1024}\right\} + E\left\{\frac{X}{2048}\right\}$$

$$C(2) = 128\,E\left\{\frac{X - 2048\,E\left\{\frac{X}{2048}\right\}}{1024}\right\} + E\left\{\frac{Y-512\left\{\frac{Y}{519}\right\}}{4}\right\}$$

$$R(2) = 64\left(Y - 4\,E\left\{\frac{Y}{4}\right\}\right) + E\left\{\frac{X - 1024\,E\left\{\frac{X}{1024}\right\}}{16}\right\}$$

Rangées d'un bloc de 16 mémoires / colonnes d'un bloc de 16 mémoires	0	1	63	64	255
0	x = 0 à 15 y = 0	x = 16 à 31 y = 0	1008 à 1023 y = 0	0 à 15 y = 1	1008 à 1023 y = 3
1	x = 0 à 15 y = 4	x = 16 à 31 y = 4	1008 à 1023 y = 4	0 à 15 y = 5	1008 à 1023 y = 7
127	x = 0 à 15 y = 508	x = 16 à 31 y = 508	1008 à 1023 y = 508	0 à 15 y = 509	1008 à 1023 y = 511
128	x = 1024 à 1039 y = 0	x = 1040 à 1053 y = 0	2032 à 2047 y = 0	1024 à 1039 y = 1	2032 à 2047 y = 3
129	x = 1024 à 1039 y = 4	x = 1040 à 1053 y = 4	2032 à 2047 y = 4	1024 à 1039 y = 5	2032 à 2047 y = 7
255	x = 1024 à 1039 y = 508	x = 2030 à 2047 y = 508	2032 à 2047 y = 508	1024 à 1039 y = 509	2032 à 2047 y = 511

d'après les tableaux ci-dessus, les formules donnant L, R, C sont donc :

$$L(2) = 4\,E\left\{\frac{Y}{1024}\right\} + E\left\{\frac{X}{2048}\right\}$$

$$C(2) = 128\,E\left\{\frac{X - 2048\,E\left\{\frac{X}{2048}\right\}}{1024}\right\} + E\left\{\frac{Y-512\left\{\frac{Y}{519}\right\}}{4}\right\}$$

$$R\quad(2) = 64\,(Y - 4\,E\left\{\frac{Y}{4}\right\}) + E\left\{\frac{X - 1024\,E\left\{\frac{X}{1024}\right\}}{16}\right\}$$

4 zones : il est composé de 16 lots de 4 zones. L'affectation des points dans les lots de 4 blocs est la suivante :

Lot 0

C	zone 0	zone 1	zone 2	zone 3
255	y = 1024 à 1279	y = 1280 à 1535	y = 1536 à 1791	y = 1792 à 2047
192	x = 1024 à 2047	x = 1024 à 2047	x = 1024 à 2047	x = 1024 à 2047
191	y = 1024 à 1079	y = 1280 à 1535	y = 1536 à 1791	y = 1792 à 2047
128	x = 0 à 1023	x = 0 à 1023	x = 0 à 1023	x = 0 à 1023
127	y = 0 à 255	y = 256 à 511	y = 512 à 767	y = 768 à 1023
64	x = 1024 à 2047	x = 1024 à 2047	x = 1024 à 2047	x = 1024 à 2047
63	y = 0 à 255	y = 256 à 511	y = 512 à 767	y = 768 à 1023
0	x = 0 à 1023	x = 0 à 1023	x = 0 à 1023	x = 0 à 1023

pour les autres lots, on procède de la même façon qu'on a procédé dans le cas de 2 zones.

Lot 12	Lot 13	Lot 14	Lot 15
Lot 8	Lot 9	Lot 10	Lot 11
Lot 4	Lot 5	Lot 6	Lot 7
Lot 0	Lot 1	Lot 2	Lot 3

0 2048 4086 6144 8181

Affectation des lots en fonction de (X,Y) pour z = 4

Le tableau ci-après donne par exemple le rangement des points dans le lot 0 et la zone 1.

Rangées d'un bloc de 16 mémoires / Colonnes d'un bloc de 16 mémoires		0	1	63	64	255
0	X	0 à 15	16 à 31	1008 à 1023	0 à 15	1008 à 1023
	Y	256	256	256	257	259
1		0 à 15	16 à 31	1008 à 1023	0 à 15	1008 à 1023
		260	260	260	261	263
63		0 à 15	16 à 31	1008 à 1023	0 à 15	1008 à 1023
		508	508	508	509	511
64		1024 à 1039	1040 à 1055	2032 à 2047	1024 à 1039	2032 à 2047
		256	256	256	257	259
127		1024 à 1039	1040 à 1055	2032 à 2047	1024 à 1039	2032 à 2047
		508	508	508	509	511
128		0 à 15	16 à 31	1008 à 1023	0 à 15	1008 à 1023
		1280	1280	1281	1283	
191		0 à 15	16 à 31	1008 à 1023	0 à 15	1008 à 1023
		2044	2044	2044	2045	2047
192		1024 à 1039	1040 à 1055	2032 à 2047	1024 à 1039	2032 à 2047
		1280	1280	1280	1281	1283
255		1024 à 1039	1040 à 1055	2032 à 2047	1024 à 1039	2032 à 2047
		2044	2044	2044	2045	2047

D'après les tableaux on a donc :

$$L(4) = 4\,E\left\{\frac{Y}{2048}\right\} + E\left\{\frac{X}{2048}\right\}$$

$$C(4) = 64\,E\left\{\frac{X - 2048\,E\left\{\frac{X}{2048}\right\}}{1024}\right\} + E\left\{\frac{Y - 256\left\{\frac{Y}{256}\right\}}{4}\right\} + 128\,E\left\{\frac{Y - 2048\,E\left\{\frac{Y}{2048}\right\}}{1024}\right\}$$

$$R(4) = 64\left(Y - 4\,E\left\{\frac{Y}{4}\right\}\right) + E\left\{\frac{X - 1024\,E\left\{\frac{X}{1024}\right\}}{16}\right\}$$

Si 1 étant le nombre total de blocs formant la mémoire d'image. Pour deux zones on divise 1 par deux, et à chaque fois que l'on augmente le nombre de zones par puissance de deux, alors on diminue le nombre de lots de façon à avoir nb de lots/zone x nb de zones = cte = 1. La limite étant donc atteinte lorsque le nombre de zones est égal au nombre total de blocs formant la mémoire d'image.

Toute l'étude a été basée pour 1 = 64.

En procédant au même procédé de rangement comme en 2 zones et 4 zones on a abouti aux expressions :

$$L(z) = a(z)\,E\left\{\frac{Y}{c(z)}\right\} + b(z)\,E\left\{\frac{X}{d(z)}\right\}$$

$$R_E(z) = C\left(Y - yE\left\{\frac{Y}{y}\right\}\right) + E\left\{\frac{X}{b}\right\}$$

$$C_E(z) = \frac{256}{z}\,E\left\{\frac{X - \alpha(z)\,E\left\{\frac{X}{\alpha(z)}\right\}}{Dv}\right\} + E\left\{\frac{Y - \frac{Dv}{z}\,E\left\{\frac{zY}{Dv}\right\}}{y}\right\} + \beta(z)E\left\{\frac{Y - \gamma(z)E\left\{\frac{Y}{(z)}\right\}}{Dv}\right\}$$

où
- b = nombre de boîtiers devant être lus ensemble
- c = nombre de rangées par ligne
- Dv = définition verticale
- y = nombre de ligne par colonne
- r = nombre total de rangées d'un boîtier mémoire.

z	1	2	4	8	16	32	64
$\alpha(z)$	Dv	2 Dv	2 Dv	4 Dv	4 Dv	8 Dv	8 Dv
$\beta(z)$	r	r	$\frac{r}{2}$	$\frac{r}{2}$	$\frac{r}{4}$	$\frac{r}{4}$	$\frac{r}{8}$
$\gamma(z)$	Dv	Dv	2 Dv	2 Dv	4 Dv	4 Dv	8 Dv
a(z)		4	4	1	1	1	
b(z)		Dv	2 Dv	4 Dv	4 Dv	4 Dv	
c(z)		1	1	2	2	0	
d(z)		2 Dv	2 Dv	2 Dv	2 Dv		

pour z = 64 on n'a pas besoin de déterminer car on n'a qu'un seul lot.

$\beta(z)$ est le nombre total de colonnes occupées pour $x \in [0,8191]$ dans une zone zi.

Le calcul de R et C a été de façon à ce qu'un calculateur puisse être mis dans n'importe quelle zone, et pour faciliter le rafraîchissement, car lorsqu'on fait une opération de lecture ou d'écriture sur une rangée,toute la rangée est rafraîchie.

Quand on a des mémoires dont le nombre de colonnes et de rangées est inférieur à 256 x 256, il faut disposer ces mémoires de façon à se ramener à 256 x 256. La raison du choix de cette disposition est telle que 1024 x 1024 = 16 x 256 x 256.

Soient :

. mode de travail en image
. Dy la définition horizontale
. z le nombre de zones
. I le numéro de l'image

le rangement des points dans la mémoire est fonction de l'organisation des boîtiers mémoire lui-même donc fonction du nombre total de colonnes et de rangées. Le choix des mémoires étant porté sur les mémoires 64 K RAM dynamiques qui ont 256 colonnes et 256 rangées. Le rangement est donné par le tableau ci-dessous, pour le mode image.

Rangées d'un bloc de 16 mémoires / Colonnes d'un bloc de 16 mémoires	0	1	63	64	255	
0	x = 0 à 15 y = 0	x = 16 à 31 y = 0	x = 1008 à 1023 y = 0	x = 0 à 15 y = 1	x = 1008 à 1023 y = 3	image n° 0
1	idem pour x y = 4	y = 4	y = 4	y = 5	y = 7	
$\frac{256}{\text{nb d'image}} - 1$					$y = \frac{Dy}{z} - 1$	
$\frac{256}{\text{nb d'image}}$	x = 0 à 15 y = 0				n = 1008 à 1023 y = 7	image n° 1
$\frac{I.\ 256}{\text{nb d'image}} - 1$						image n° i
$\frac{I.256}{\text{nb d'image}}$						image n° i+1
$255 - \frac{256}{\text{nb d'image}}$	x = 0 à 15 y = 0	x = 16 à 31 y = 0	x = 1008 à 1023 y = 0	x = 0 à 13 y = 1	x = 1008 à 1025 y = 3	
						dernière image

Rangées d'un bloc de 16 mémoires / Colonnes d'un bloc de 16 mémoires	0	1	63	64	255	
0	x = 0 à 15 y = 0	x = 16 à 31 y = 0	x = 1008 à 1023 y = 0	x = 0 à 15 y = 1	x = 1008 à 1023 y = 3	image n° 0
1	idem pour x y = 4	y = 4	y = 4	y = 5	y = 7	
$\frac{256}{\text{nb d'image}} - 1$					$y = \frac{Dy}{z} - 1$	
$\frac{256}{\text{nb d'image}}$	x = 0 à 15 y = 0				n = 1008 à 1023 y = 7	image n° 1
$I.\frac{256}{\text{nb d'image}} - 1$						image n° i
$I.\frac{256}{\text{nb d'image}}$						image n° i+1
$255 - \frac{256}{\text{nb d'image}}$	x = 0 à 15 y = 0	x = 16 à 31 y = 0	x = 1008 à 1023 y = 0	x = 0 à 13 y = 1	x = 1008 à 1025 y = 3	
						dernière image

1.3.2. Calcul de l'adresse rangée R : en fonction du tableau précédent, on a :

$$R_I = C\left(Y - y\ E\left\{\frac{Y}{y}\right\}\right) + E\left\{\frac{X}{b}\right\}$$

avec E fonction partie entière
C nombre de rangées par ligne
y nombre de lignes sur toutes les 256 rangées.

1.3.3. Calcul de l'adresse colonne C : en fonction du même tableau, on a :

$$C_I(z) = \frac{256}{z}\left(I - z\ E\left\{\frac{I}{z}\right\}\right) + E\left\{\frac{Y - \frac{Dv}{z}\ E\left\{\frac{zY}{Dv}\right\}}{y}\right\}$$

I numéro de l'image

Calcul du boîtier mémoire : il est le même que celui annoncé dans la partie principes généraux d'un système graphique à balayage de trame.

3.3. Description du circuit d'aiguillage -

Le circuit d'aiguillage est formé par l'ensemble :

- tampons récepteurs
- circuit de décodage
- réseau d'interconnexion
- tampons émetteurs

Les ordonnées yi des informations provenant des tampons récepteurs sont décodées pour former la destination Di. Suivant les destinations Di, le réseau d'interconnexion range les informations dans les tampons émetteurs des zones correspondantes à Di.

Du côté réception, le dialogue entre calculateurs et tampons récepteurs se fait avec les signaux :

- autorisation d'écriture
- indicateur de plein
- occupé

Un calculateur ne peut écrire dans un tampon que s'il est libre, pour cela il active le signal d'autorisation d'écriture et envoie les données sur le bus de données.

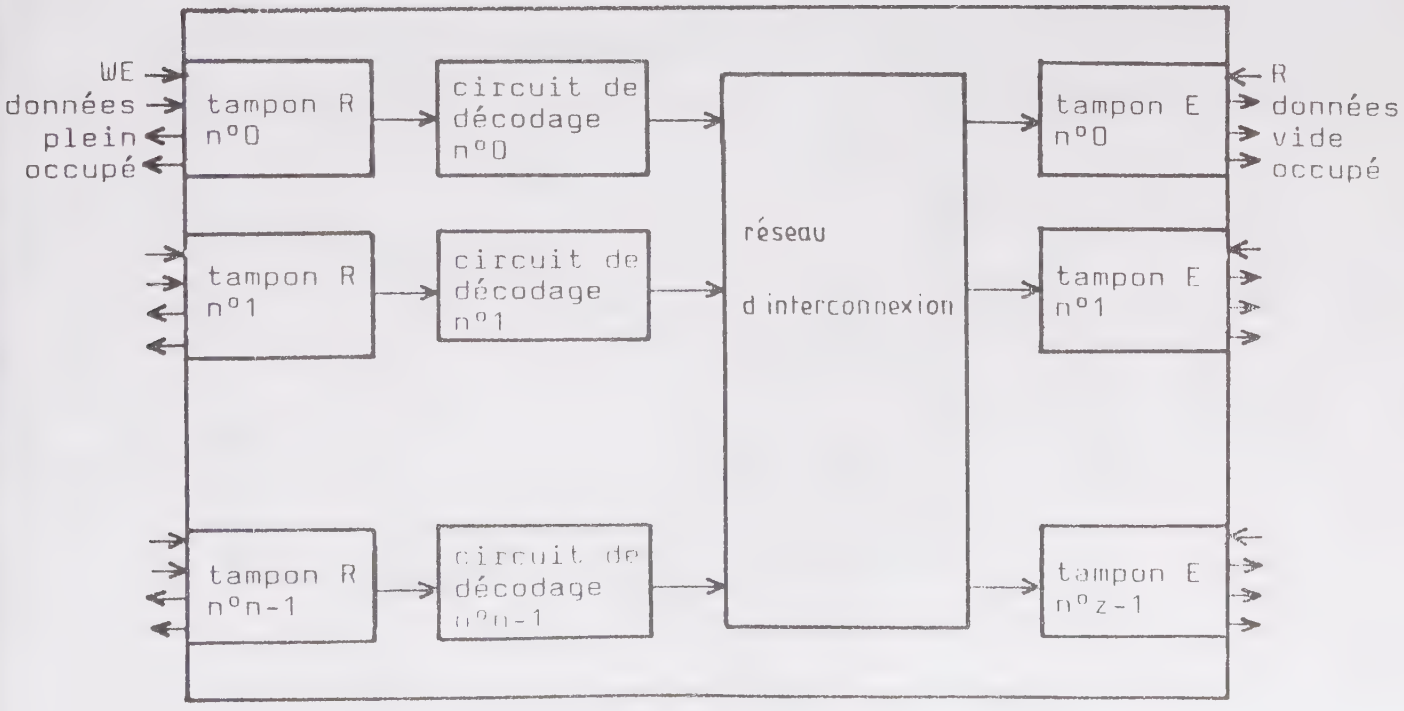
Quand le signal indicateur de plein est actif alors le calculateur correspondant doit s'arrêter.

Dès que le signal plein n'est plus actif, le calculateur peut redémarrer.

Du côté émission, le dialogue entre circuits de mise à jour de l'image et tampons émetteur se fait avec les signaux :

- autorisation de lecture
- indicateur de vide
- occupé.

Un circuit de mise à jour de l'image ne peut acquérir les données que si le tampon émetteur correspondant est libre (signal occupé inactif), dans ce cas il active le signal d'autorisation de lecture et prend les données. Quand le signal indicateur de vide est actif, le circuit de mise à jour correspondant est arrêté.



3.4. Description du circuit de mise à jour de l'image -

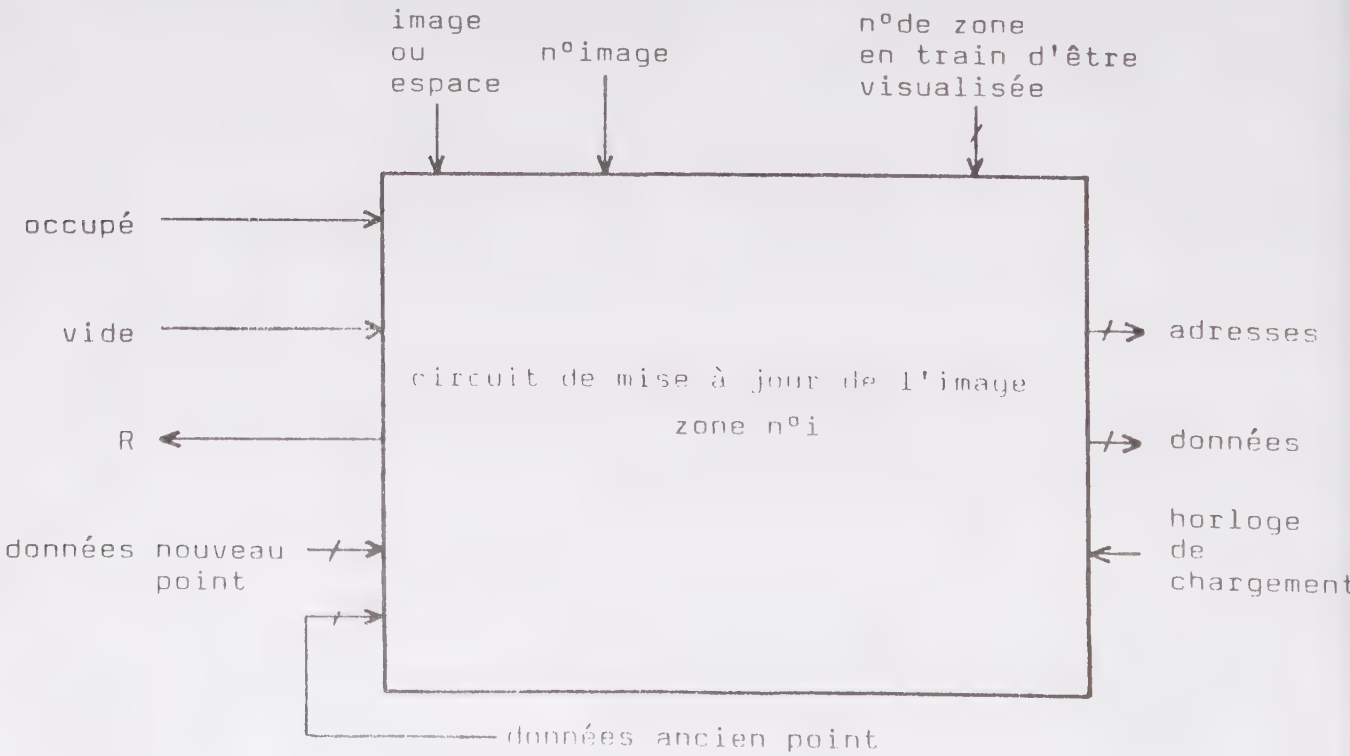
Il s'occupe du chargement de la mémoire d'image en fonction de paramètres statiques et dynamiques.

Les paramètres statiques sont :

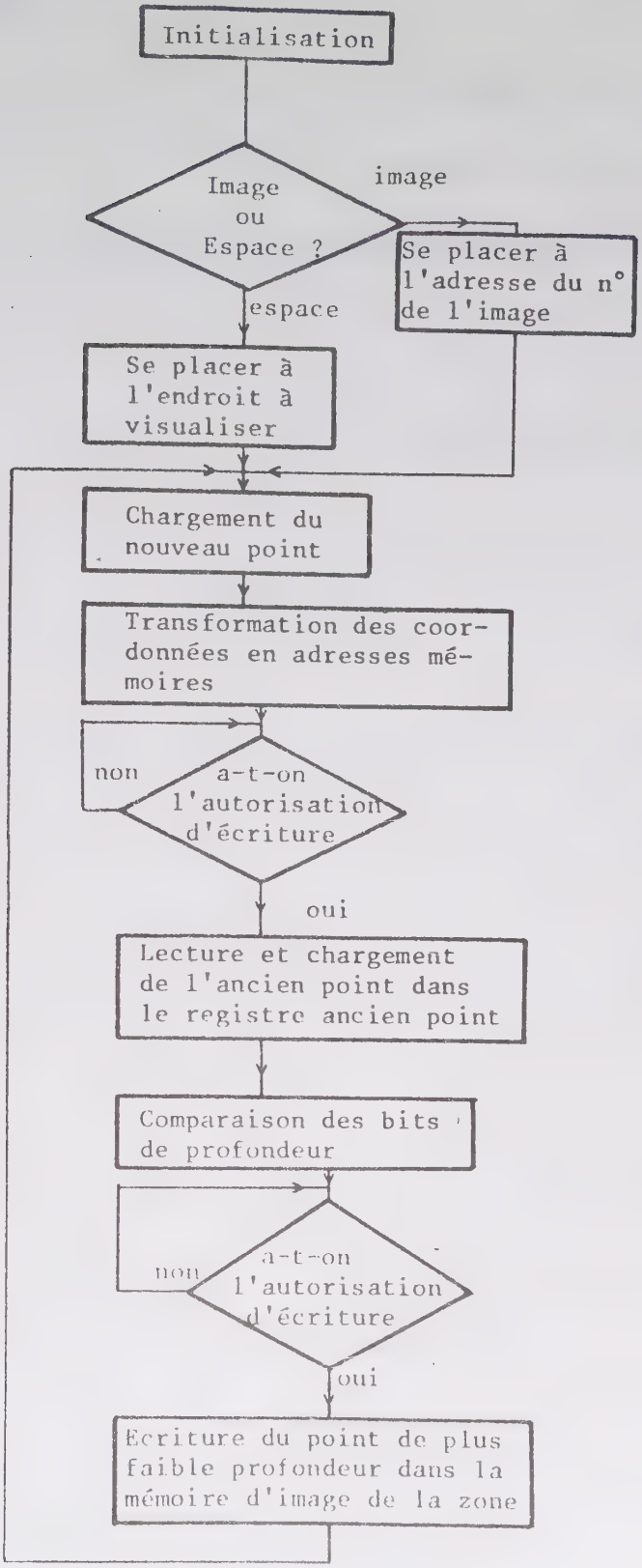
- travailler avec ou sans fenêtre de visualisation dans un espace adressable,
- travailler avec plusieurs images différentes,
- No d'images à visualiser et à charger,
- coordonnées de l'espace à visualiser.

Les paramètres dynamiques sont :

- les états de la zone (cycle d'écriture, cycle de rafraîchissement des mémoires, cycle de visualisation),
- les coordonnées (x,y), la profondeur et la couleur des points.



3.4.1. Organigramme de fonctionnement du circuit de mise à jour -



L'initialisation se fait par la commande RESET.

3.4.2. Fonction interne -

Rafraîchissement des mémoires de la zone.

Pendant la période de visualisation, le circuit de mise à jour de l'image procède à une lecture de toute la zone à la cadence de l'horloge de chargement.

3.5. Description du séquenceur -

Son rôle est d'envoyer à tous les circuits de mise à jour de l'image, le numéro de zone en train d'être visualisée.

4. ORGANE VIDEO -

Il se compose d'un circuit de rafraîchissement de l'image et d'un générateur de signaux de synchronisation.

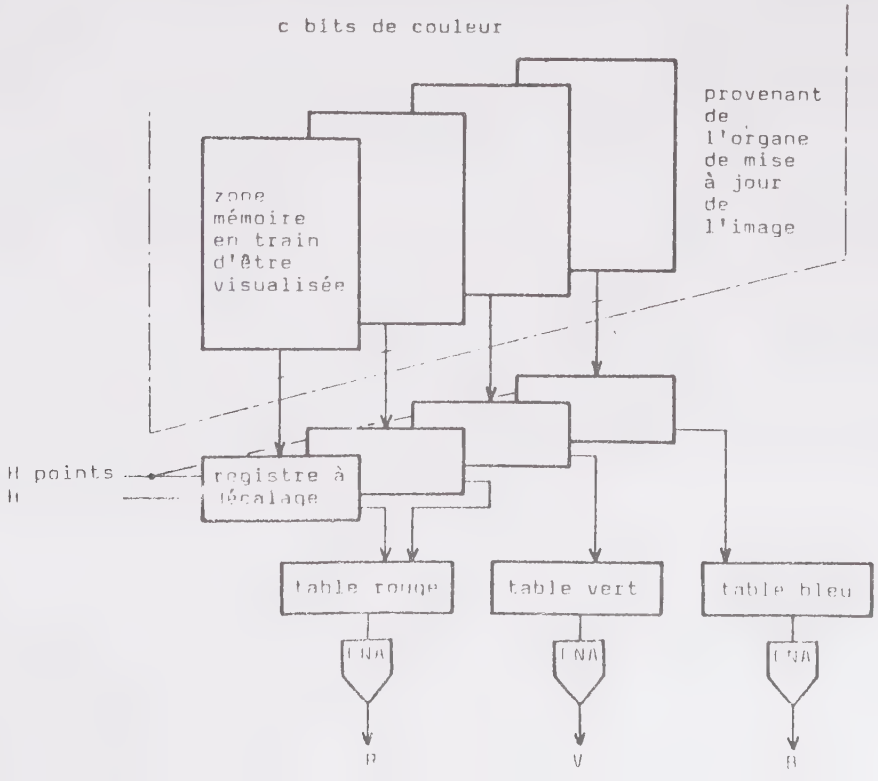
4.1. Description du circuit de rafraîchissement de l'image -

Seules, les informations couleurs sont prises en compte. Les informations couleurs relatives à chaque point provenant de chaque zone sont transformées sous forme de signaux vidéo R.V.B. en synchronisme avec les signaux de synchronisation verticaux et horizontaux.

L'information couleur de C bits est divisée en trois qui vont chacune dans une table de couleurs correspondante, ce qui permet de définir une échelle relative de niveau d'un primaire par rapport à son échelle absolue.

Pour chaque primaire (R ou V ou B), l'échelle absolue est définie par le nombre de bits de sortie de la table de couleurs correspondante dont le contenu peut être varié au gré de l'utilisateur. L'échelle relative est définie par les $\frac{C}{3}$ bits de couleurs correspondants à la primaire.

Le fait de varier le contenu de la table des couleurs permet de faire translater l'échelle relative sur l'échelle absolue pour chaque primaire (R,V,B).



Les tables de couleurs sont des mémoires vives (RAM) très rapides. Elles peuvent être chargées à tout moment à la seule condition de les charger en dehors de la période de visualisation globale.

4.2. Description du générateur de signaux de synchronisation -

Il fournit tous les signaux d'horloge nécessaires au fonctionnement du système et les signaux de synchronisation avec le moniteur télévision.

Les signaux d'horloge sont :

- signal de rafraîchissement HRFSH
- horloge de validation des colonnes rangées HRAS

Les signaux de synchronisation sont :

- synchronisation verticale
- synchronisation horizontale
- suppression ligne
- suppression trame.

L'horloge de base étant l'horloge de décalage des points.

CHAPITRE IV

SYNTHESE DES DIFFERENTS ORGANES

A.ORGANE DE MISE A JOUR DE L'IMAGE

1.-Conception du circuit d'aiguillage:

1.1.-Conception des tampons:

Les fonctions des tampons sont identiques.Ainsi le tampon émetteur est le même que le tampon récepteur.

Un tampon est fait à base de mémoire vive (RAM). Il se compose d'un circuit de contrôle de la mémoire vive. Il y a trois phases différentes: phase d'écriture,phase de lecture et phase bloquée (on ne fait rien).

Le principe d'adressage de la mémoire vive est la suivante:à t=0,la mémoire vive est supposée vide. Le signal vide est donc actif. Dans ce cas , aucune lecture de la mémoire ne peut se faire. Mais,dès qu'on reconnaît le d'autorisation d'écriture ($\overline{WE}$ actif),on fait une écriture dans la mémoire vive,l'adresse écriture est incrémentée d'une unité et le signal vide n'est plus activé. Dans ce cas une lecture de la mémoire vive peut se faire. Dans le cas général où t≠0 l'adresse écriture se trouve à l'adresse"e" et l'adresse lecture à l'adresse "l".

Supposons maintenant qu'on est à l'adresse "l" et à l'adresse "e". Après chaque cycle de lecture,l'adresse lecture est incrémentée et après chaque cycle d'écriture l'adresse écriture est aussi incrémentée. La mémoire vive fonctionne comme un "FIFO" (First In First Out).

Pour ceci on dispose donc d'un compteur d'adresse écriture qui est incrémenté après chaque cycle d'écriture et bloqué dans les autres cas,et d'un compteur d'adresse lecture dont le principe est le même après chaque cycle de lecture.

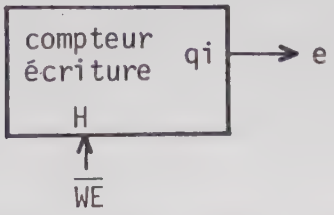
Pour adresser la mémoire vive,les deux adresses "e" et "l" sont multiplexées. Pendant le cycle d'écriture ($\overline{WE}$ actif),l'adresse de la mémoire vive est l'adresse écriture sinon c'est l'adresse lecture.

Le tampon est vide après un cycle de lecture si l'adresse lecture est identique à l'adresse écriture. Dans ce cas,le signal vide est activé. Tout cela est caractérisé par "e=l" et $\overline{WE}$ non actif : alors sur le front descendant de "R" le signal vide est activé.

Le tampon est plein lorsqu'une écriture a été faite à l'adresse écriture juste avant l'adresse lecture. Ceci est caractérisé par: "e=l-l" ou "e+l=l et e<l" et par R non actif,alors le signal plein est activé sur le front montant de $\overline{WE}$.

1.1.1-Compteur écriture:

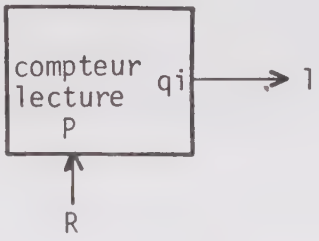
Il n'est incrémenté que sur le front montant de $\overline{WE}$,dans les autres cas il reste bloqué à l'adresse présente. Sa taille est égale à la taille de la mémoire vive. Dès qu'il a dépassé sa taille maximale,il se remet automatiquement à zéro.



Le signal d'horloge est $\overline{WE}$
La sortie est "e"

1.1.2.-Compteur lecture:

Il n'est incrémenté que sur le front descendant de R. Dans les autres cas il reste à l'adresse présente. Sa taille est aussi égale à celle de la mémoire vive. Dans le cas où celle-ci est vide (signal v=1),le compteur est bloqué.



Le signal d'horloge est R
La sortie est "1"

1.1.3.-Multiplexeur:

En cycle d'écriture $\overline{WE}$=0,la sortie "a" est l'adresse "e". Dans les autres cas ($\overline{WE}$=1),la sortie "a" est l'adresse "1".

"a" a pour équation logique:

$$a=WE.e+\overline{WE}.1$$

1.1.4.-Réalisation du signal vide:

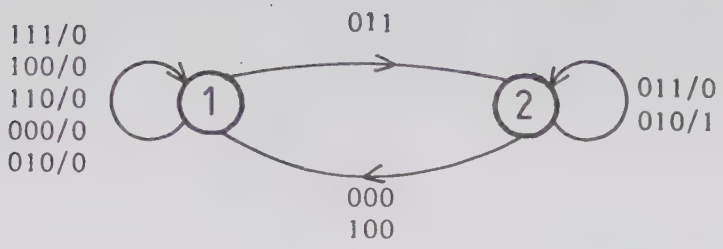
Les adresses "e" et "1" sont comparées à l'aide d'un comparateur. Si "e=1",la sortie "x" du comparateur est à "0" et si "e≠1","x=1". Quand " e≠1",c-à-d "x=1",le signal vide "v" doit rester à "0". Il passe à "1" sur le front descendant de R quand "x=0" et $\overline{WE}$=1 (c'est à dire qu'on vient de faire la lecture de la dernière information contenue dans la mémoire vive), le signal vide reste dans cet état tant qu'on n'a fait aucune écriture dans la mémoire vive. Dès qu' une écriture commence à se faire (front descendant

de $\overline{WE}$),le signal vide "v" est immédiatement mis à "0".

Le graphe de fluence est le suivant:- les entrées sont x,$\overline{WE}$,R

- la sortie est v

$x\overline{WE}R$ / v



Les équations logiques sont:

$y = R\ \overline{x}\ \overline{Y} + \overline{WE}\ Y$ pour la variable secondaire

$J = R\ \overline{x}$ avec une bascule JK

$K = WE$

$v = \overline{R}\ \overline{WE}\ Y$

1.1.5.-Réalisation du signal plein:

Le principe est le même que pour le signal vide,sauf qu'on compare "e+1" à "1" et que l'activation du signal plein "p" se fait sur le front montant de $\overline{WE}$. Quand on est dans l'état plein "p=1","p" ne repasse à "0" que lorsqu' on fait une lecture. Le passage de "p=1" à "p=0" se fait sur le front descendant de R. Quand la mémoire vive n'est pas pleine (e+1≠1),"p" reste toujours à "0" que l'on fasse une lecture,une écriture ou rien.

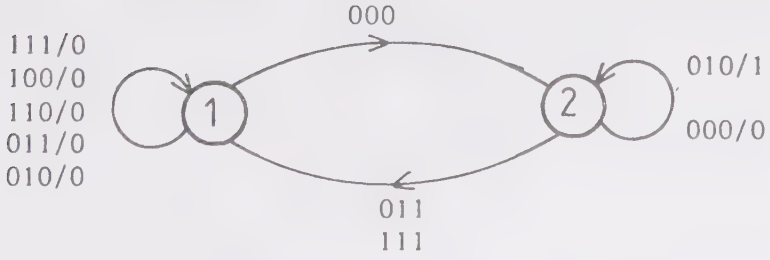
Quand "e+1=1",si on fait une lecture, "p" reste à "0". De même si on ne fait rien. Mais dès que l'on fait une écriture,sur le front montant de $\overline{WE}$ (fin d'écriture),"p" est activé (état 1).

La comparaison de "e+1" à "1" donne le signal "t" qui vaut "0" quand "e+1"="1" et "1" quand "e+1"≠"1".

Le graphe de fluence est le suivant:- les entrées sont t,$\overline{WE}$,R

- la sortie est p

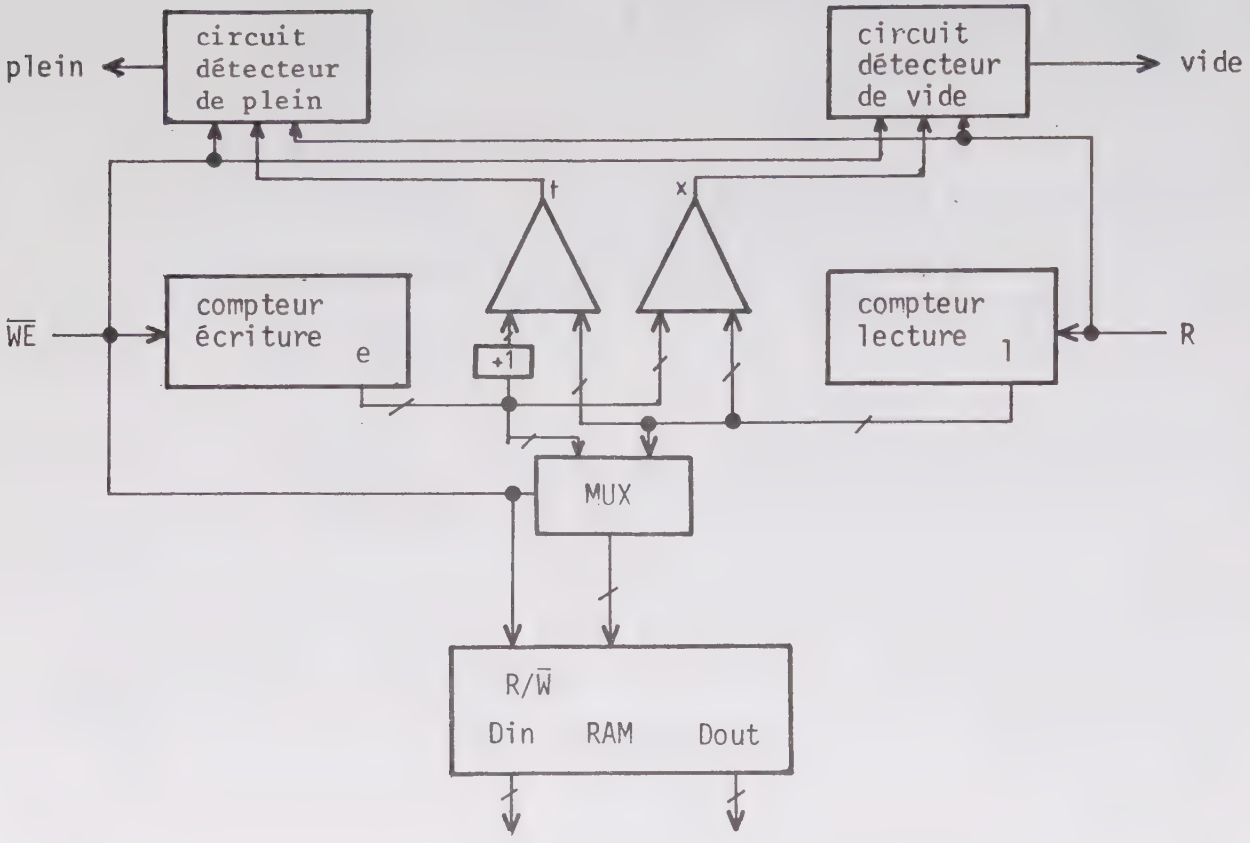
t $\overline{WE}$ R / p



Les équations logiques sont:

$y = \bar{t}\ WE\ \bar{Y} + \bar{R}\ Y$	pour la variable secondaire
$J = \bar{t}\ WE$	avec une bascule JK
$K = R$	
$p = \bar{R}\ WE\ Y$	pour la sortie

1.1.6. -Organisation d'un tampon:



1.2.-Conception du réseau d'interconnexion:

1.2.1.-Problème:

Il s'agit de connecter "n" entrées à "z" sorties.

Ceci peut être fait à l'aide d'un " réseau delta " , utilisant des blocs élémentaires "axb" interconnectant "a" entrées à "b" sorties.

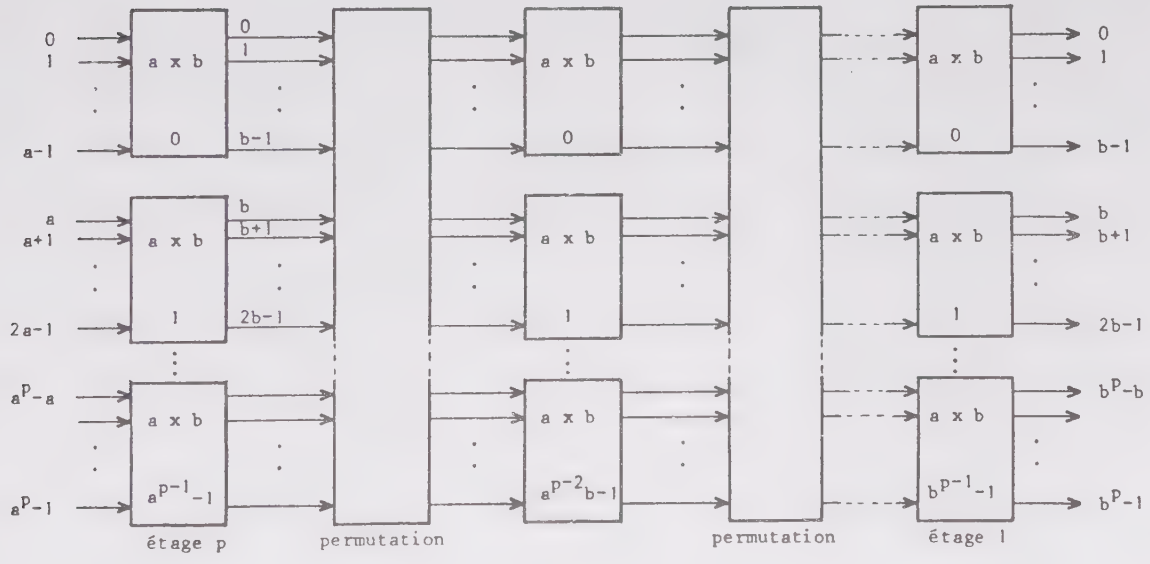
Un " réseau delta " ayant "a^p" entrées et "b^p" sorties comporte "p" étages de réseaux élémentaires "axb". Chaque étage "e" comporte " $a^{e-1}b^{p-e}$ " réseaux élémentaires. La liaison entre étage se fait par connexion statique obtenue par permutation de fils. La loi de permutation étant la suivante:

$$p(i) = ai \bmod(m-1) \quad \text{si } i \neq m-1$$
$$= i \quad \text{si } i = m-1$$

avec: .i est le numéro des sorties de l'étage précédent et p(i) le numéro de l'entrée de l'étage suivant auquel la sortie "i" doit être connectée.

.m est le nombre d'entrées de l'étage suivant. Il est égal au nombre de sorties de l'étage précédent.

.a est le nombre d'entrées d'un réseau élémentaire.



Le nombre de réseaux élémentaires "a x b" est:

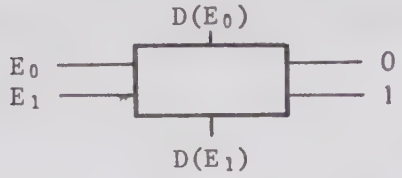
$$\sum_{1\leq i\leq p} a^{p-i}b^{i-1} = \frac{a^p-b^p}{a-b} \quad \text{si } a\neq b$$
$$= pb^{p-1} \quad \text{si } a=b$$

On montre que si $n = a^p$ et $z = b^p$ alors un réseau d'interconnection en delta à "p" étages convient.

Dans notre étude nous avons choisi des blocs élémentaires "2 x 2" que nous appellerons "aiguilleurs". De ce fait le nombre de calculateurs possibles serait de "2^p",le nombre de zones mémoires "2^q".

Description succinte d'un aiguilleur:

Les entrées sont (E_0,E_1). Les sorties sont codées en binaire par "0" et "1". Les commandes sont les destinations,$D(E_0)$ et $D(E_1)$.



Si $D(E_0) = 0$ l'entrée E_0 est connectée à la sortie 0

$D(E_0) = 1$ ——————————"——————————— 1

Si $D(E_1) = 0$ l'entrée E_1 est connectée à la sortie 0

$D(E_1) = 1$ ——————————"——————————— 1

Etats d'un aiguilleur:

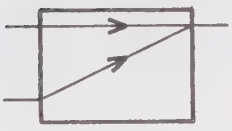
Un aiguilleur a quatre états possibles suivant les destinations d_0i et d_1i.

d_0 est le bit de $D(E_0)$

d_1 est le bit de $D(E_1)$

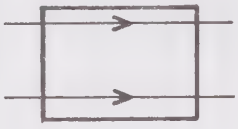
avec conflit

d_0i d_1i=00

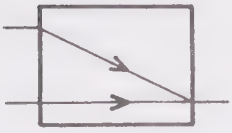


sans conflit

d_0i d_1i=01



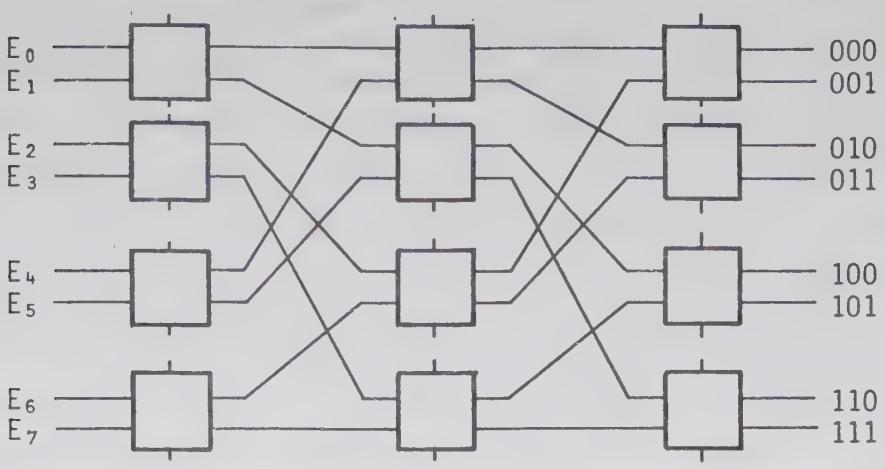
d_0i d_1i=11



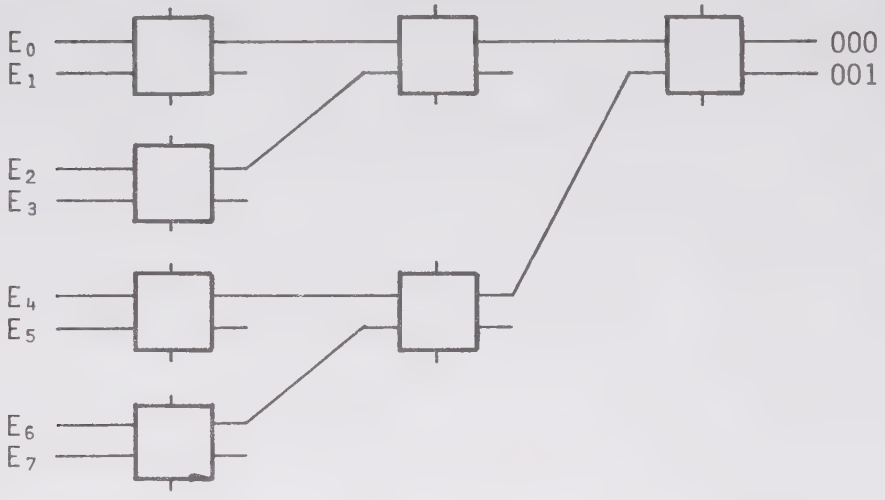
d_0i d_1i=10

Exemples de configuration:

. le nombre de calculateurs est égal au nombre de zones mémoires. Ce nombre est égal à huit.



. le nombre de calculateurs est égal à huit
le nombre de zones mémoires est égal à deux.



1.2.2.-Fonctionnement:

On considère un réseau d' interconnexion 2^3x2^3 . La destination est codée en binaire par les bits d_2 d_1 d_0.

Dans le mot de destination d_2 d_1 d_0 : au niveau de l'étage 3,seule d_2 est prise en compte;à l'étage 2 c'est d_1 ,et à l'étage 1 c'est d_0.

Au niveau de l'étage 3,on a un décodeur qui regarde l'état du tampon récepteur: s'il n'est ni vide,ni en période d'écriture ,le décodeur fait l'acquisition des données points par lecture du tampon,détermine la destination et met les données points et destination dans un registre interne. Dès que le transfert de ces données est fait,il envoie une demande de passage à l'aiguilleur correspondant. La demande est maintenue tant qu'on n'a pas reçu l'autorisation de passage. Dès que le décodeur a reçu l'autorisation de passage,c'est l'aiguilleur de l'étage 3 qui prend en charge les données. Tant que les données ne sont pas transférées dans l'un des deux registres de sortie de l'aiguilleur de l'étage correspondant,le décodeur ne peut prendre d'autres informations.

Le principe de la prise en charge des données est le même pour tout les aiguilleurs. Dès que les données sont transférées dans l'un des deux registres de l'étage 3,celui-ci fait une demande de passage à l'étage 2 qui lui répond par une autorisation de passage ou non. La demande de l'aiguilleur de l'étage 3 est maintenue tant que les données ne sont pas transférées à l'un des registres de sorties de l'aiguilleur demandé de l'étage 2. Dès que le transfert est fait,le destinataire de l'étage 2 fait une demande de passage à l'étage 1 qui lui répond par le même principe énoncé précédemment. Dès que les données arrivent à l'un des registres de sortie de l'étage 1, celui-ci fait une demande de chargement du tampon émetteur. La demande est maintenue tant que les données ne sont pas chargées dans le tampon émetteur.

Quand le tampon récepteur est vide,on est en période d'écriture: l'étage 3 ne reçoit aucune demande de passage.

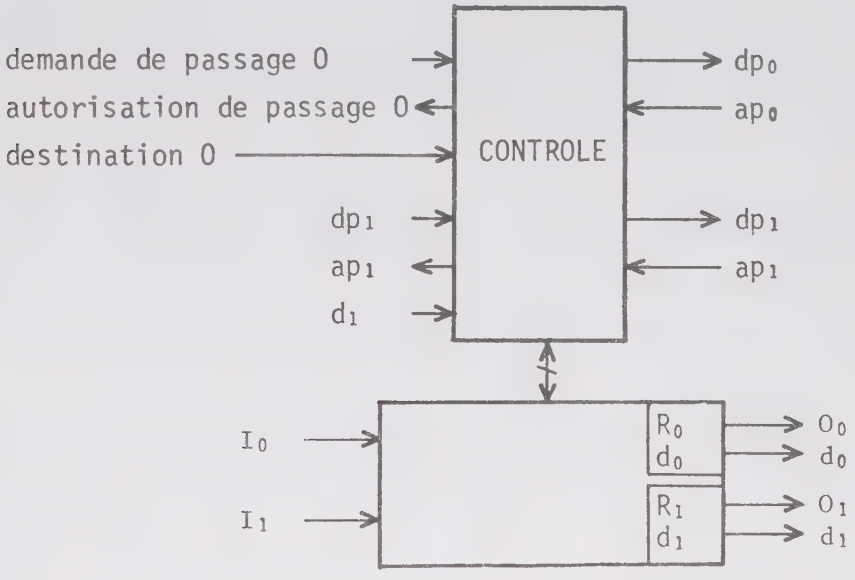
En cas de conflit (c-à-d deux demandes de passage arrivant simultanément pour une même destination) une règle de priorité s'impose. La demande la plus prioritaire est celle qui provient de la voie correspondant à la voie demandée.

1.2.3.-Conception de l'aiguilleur:

Suivant les demandes et l'état des registres internes (vides ou pleins) il envoie les signaux d'autorisation de passage du côté du demandeur. Quand il est demandeur,il envoie les signaux "demande de passage" dès que les données sont disponibles dans l'un ou les deux registres. En cas de conflit,il fait lui-même l'arbitrage,compte tenu de la règle de priorité énoncée précédemment.

L'aiguilleur est composé d'un circuit de contrôle et d'un échangeur,à la sortie duquel on a deux registres internes. Dans ces registres on met temporairement les informations points et les destinations,jusqu'à l'obtention de l'autorisation de passage de l'aiguilleur de l'étage suivant.

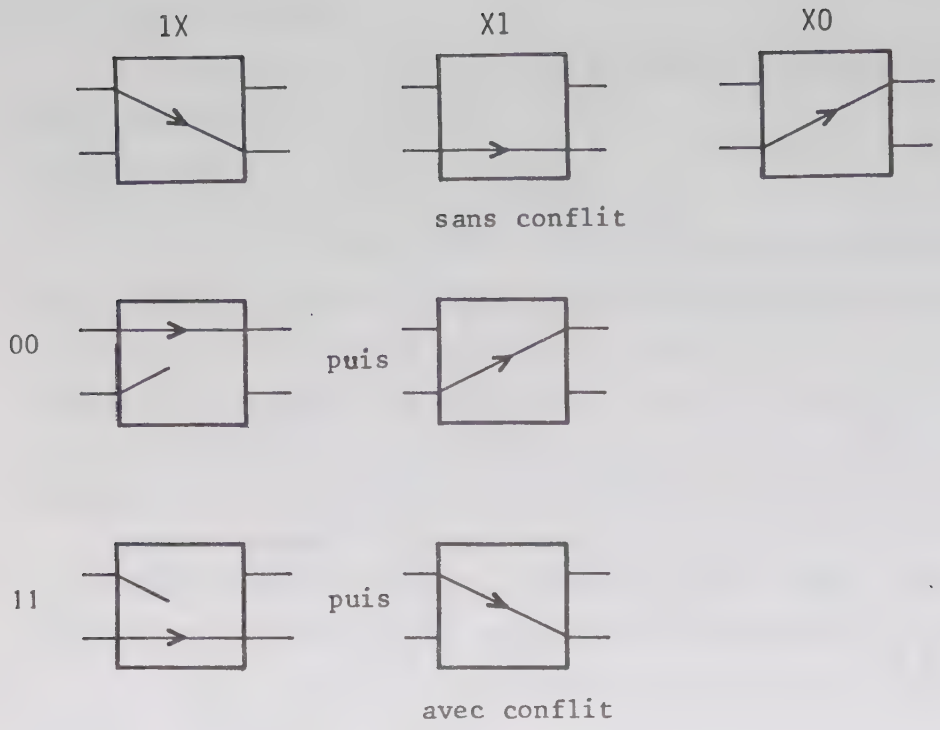
Suivant la rangée où on veut mettre l'aiguilleur,un jeu de microinterrupteurs permet de sélectionner les deux bits correspondants à la position de cette rangée, parmi les d bits de destination.



Etats de l'échangeur:

Suivant les destinations d_0i d_1i on a les états suivants:

01 10 0X



Elaboration des signaux "autorisation de passage":actifs à l'état haut

Une autorisation de passage ne peut être active que si l'on a reçu la demande de passage correspondante.

Dès qu'on a une demande de passage, on regarde sa destination et l'état du registre interne correspondant. L'autorisation de passage correspondant est activée si le registre correspondant est libre. Sinon elle n'est pas activée,la demande étant toujours maintenue. Dès que le registre est libre, l'autorisation de passage est activée.

En cas de demande simultannée,deux cas peuvent se présenter:-les deux demandes veulent aller dans deux voies différentes. Dans ce cas il n'y a pas de conflit.

-les deux demandes veulent aller dans une même voie. Dans ce cas,on a un conflit et l'arbitrage se fait suivant la règle de priorité énoncée précédemment.

En résumé: en cas de non conflit, les autorisations de passage sont égales au produit de la demande de passage et de l'état du registre demandé.

en cas de conflit (demande simultanée sur une même direction, le registre correspondant étant libre),on regarde la destination commune: celle qui a la priorité est celle qui est sur la même voie que la destination.

Supposons que les deux sources veulent aller en même temps vers la voie du bas et que le registre du bas est vide. Alors l'autorisation de passage du bas est activée et celle du haut ne l'est pas. Après l'écriture de la source 1 du bas dans le registre 1 du bas, les deux autorisations sont inhibées ($v_1=0$).

Si le registre 1 est de nouveau vide (après avoir envoyé les données dans la prochaine destination, $v_1=1$), alors l'autre autorisation de passage est validée et dès la fin d'écriture de la donnée de la source 0 dans le registre 1, les deux autorisations de passage sont inhibées. Entre temps, après le passage de la première source, si une autre demande arrive alors, en cas de croisement ou de ligne droite, on est dans le cas de non conflit. Ou alors, en cas de second conflit, celle qui est en attente est prioritaire.

Le graphe de fluence est le suivant:

où les signaux de commande sont:

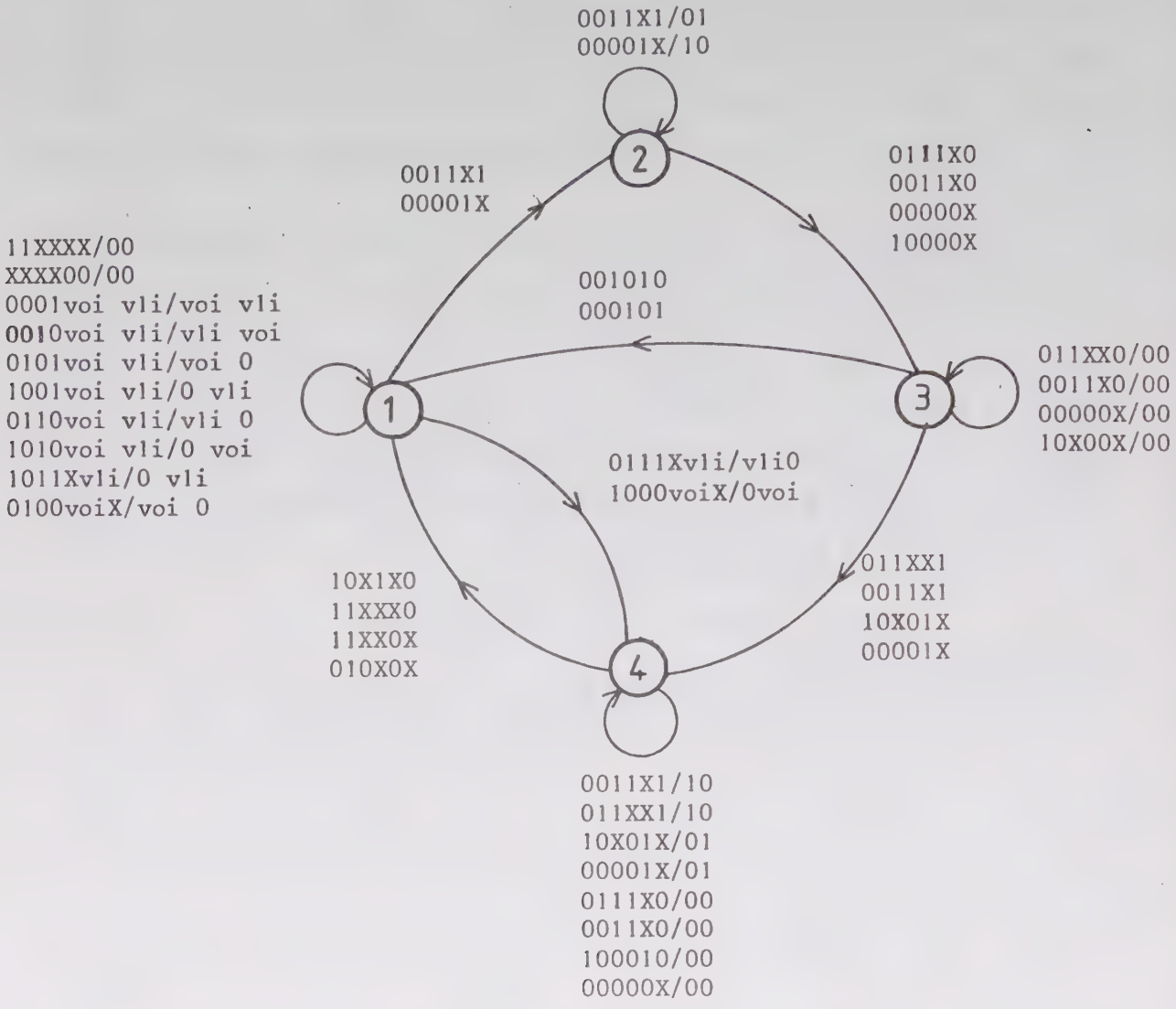
- dp_{0i+1}: demande de passage à la voie du haut provenant de l'aiguilleur de l'étage i+1.
- dp_{1i+1}: même chose mais pour la voie du bas.
- doi: destination à l'étage i provenant de la voie du haut.
- dli: " " " " bas
- voi: état du registre de sortie du haut de l'aiguilleur i
- vli: " bas "

où les signaux de sorties sont:

- ap_{0i+1}: autorisation de passage de l'étage i+1 à la voie du haut.
- ap_{1i+1}: " " " bas

les signaux de sortie sont actifs à l'état haut.

dp_{0i+1} dp_{1i+1} doi dli voi vli / ap_{0i+1} ap_{1i+1}



Les équations logiques sont:

$$y_1 = \overline{Y}_1 (\overline{X}_1\overline{X}_2X_3\overline{X}_4+X_1X_2+\overline{X}_3X_4+X_2\overline{X}_3+X_1X_4)+Y_1(X_2\overline{X}_4+\overline{Y}_2\overline{X}_2X_3+X_1X_2+\overline{X}_3X_4+\overline{X}_2X_3X_6+Y_2\overline{X}_4X_5+\overline{Y}_2\overline{X}_1\overline{X}_2$$

$$y_2=\overline{Y}_2(Y_1\overline{X}_1\overline{X}_2\overline{X}_3\overline{X}_4X_5+Y_1\overline{X}_1\overline{X}_2X_3X_4X_6)+Y_2(Y_1+X_2\overline{X}_6+X_4\overline{X}_6+\overline{X}_3\overline{X}_4\overline{X}_5+\overline{X}_5\overline{X}_6)$$

pour les variables secondaires où: $X_1X_2X_3X_4X_5X_6$ sont dp_{0i+1}, dp_{1i+1}, doi, dli, voi, vli respectivement.

Avec des bascules JK on a:

$$J_1=\overline{X}_1\overline{X}_2X_3\overline{X}_4+X_1X_2+\overline{X}_3X_4+X_2\overline{X}_3+X_1X_4$$

$$\overline{K}_1=X_2\overline{X}_4+\overline{Y}_2\overline{X}_2X_3+X_1X_2+\overline{X}_3X_4+\overline{X}_2X_3X_6+Y_2\overline{X}_4X_5+\overline{Y}_2\overline{X}_1\overline{X}_2$$

$$J_2=Y_1\overline{X}_1\overline{X}_2(\overline{X}_3\overline{X}_4X_5+X_3X_4X_6)$$

$$\overline{K}_2=X_2\overline{X}_6+Y_1+X_4\overline{X}_6+\overline{X}_3\overline{X}_4\overline{X}_5+\overline{X}_5\overline{X}_6+X_1X_3\overline{X}_5$$

pour les sorties on a:

$$ap_{0i+1}=Y_1\overline{X}_1\overline{X}_3X_5+\overline{X}_1X_2\overline{X}_3X_5+\overline{X}_1X_2X_5X_6+\overline{Y}_1\overline{X}_1X_3X_6+\overline{X}_1\overline{X}_2\overline{X}_3X_4X_5+\overline{X}_1\overline{X}_2X_3\overline{X}_4X_6+Y_1\overline{X}_1\overline{X}_2X_3\overline{X}_4$$
$$+\overline{X}_1X_2X_3X_4X_6+\overline{X}_1X_2X_3X_6$$

$$ap_{1i+1}=Y_1\overline{X}_2X_4X_6+X_1\overline{X}_2\overline{X}_4X_5+Y_1\overline{X}_2X_3\overline{X}_4X_5+\overline{Y}_1\overline{X}_2\overline{X}_3\overline{X}_4X_5+Y_2\overline{X}_3X_4$$

Elaboration des signaux "demande de passage": actifs à l'état bas

Les signaux "demande de passage" ne sont actifs que si les registres i_0 et i_1 correspondants sont chargés.

Supposons qu'à l'état initial, aucun des registres R_{0i} et R_{1i} n'est chargé. Dans ce cas aucune "demande de passage" n'est activée.

Dès que le registre R_{0i} est chargé (front montant du signal d'écriture du registre C_{0i}), le signal "demande de passage " dp_{0i} pour la voie du haut est activé. Il reste actif tant qu'on n'a pas l'autorisation de passage correspondante, et tant que le registre destinataire R_{i-1} n'est pas chargé. Dès que celui-ci est chargé, l'autorisation de passage correspondante retourne à l'état inactif (front descendant de ap_{0i}) et dp_{0i} retourne à l'état inactif, c'est-à-dire à son état initial.

Il en est de même pour le signal "demande de passage" dp_{1i} pour la voie du haut.

Dans le cas où les deux registres R_{1i} et R_{0i} sont chargés en même temps, l'établissement des signaux "demande de passage" est le même, mais il se fait de façon simultanée, le conflit étant arbitré par le circuit d'élaboration des signaux "autorisation de passage" de l'étage i-1.

Le graphe de fluence est le suivant:

où les signaux de commande sont: ap_{0i}: autorisation de passage de la voie du haut à l'étage i-1.

ap_{1i}: même chose mais pour la voie du bas

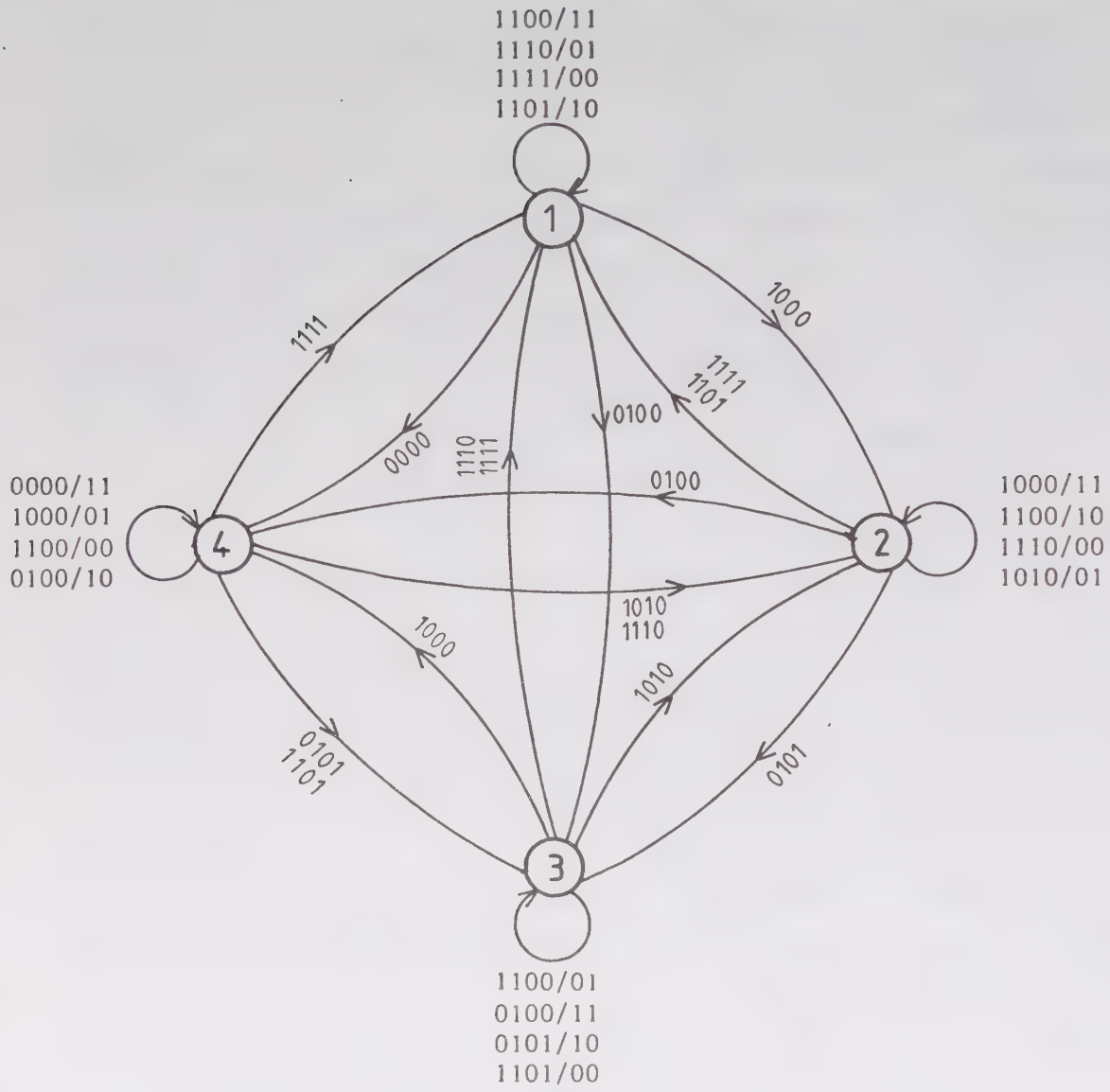
C_{0i}: écriture du registre du haut R_{0i}

C_{1i}: écriture du registre du bas R_{1i}

où les signaux de sortie sont: dp_{0i}: demande de passage de la voie du haut à l'étage i-1.

dp_{1i}: même chose mais pour la voie du bas

C_0i C_1i ap_0i ap_1i /dp_0i dp_1i



Les équations logiques sont:

$y_1=Y_1\overline{X}_3+\overline{Y}_1\overline{X}_1$ pour les variables secondaires

$y_2=Y_2\overline{X}_4+\overline{Y}_2\overline{X}_2$

$J_1=\overline{X}_1$

$\overline{K}_1=\overline{X}_3$ avec des bascules JK

$J_2=\overline{X}_2$

$\overline{K}_2=\overline{X}_4$

$dp_0i=\overline{X}_1+\overline{Y}_1\overline{Y}_2\overline{X}_3+\overline{Y}_1X_1\overline{X}_3+\overline{Y}_1\overline{X}_3X_4$ pour les sorties

$dp_1i=\overline{X}_2+\overline{Y}_2\overline{X}_4$

X_1, X_2, X_3, X_4 sont respectivement C_0i, C_1i, ap_0i, ap_1i respectivement

Circuit de chargement des registres internes R_0i et R_1i:

Principe de chargement des registres:

Dans de nombreux cas,pour diminuer l'encombrement du bus de données quand celles ci sont importantes,il faut passer ces données en plusieurs fois. Le compteur est validé par un signal "validation de comptage" Vci qui est actif dès que le signal d'écriture du registre correspondant est actif. Le signal "validation de comptage" est désactivé sur le front descendant du signal "fin de comptage" qi,qi étant le bit de plus fort poids du compteur correspondant.

Soient les étages i+1,i,et i-1. Supposons que i+1 fait une demande de passage à i. Deux cas peuvent se poser: vi=0 ou vi=1.

Si vi=0,1'autorisation de passage ap_{i+1} n'est pas activée (ap_{i+1}=0) et la demande de passage dp_{i+1} reste toujours valide (dp_{i+1}=0). Dès que le registre i-1 est vide (v_{i-1}=1),ce qui entraine ap_i=1,alors C_{i-1}=0,Vc_{i-1}=1 et Vc_i=1. Le transfert des informations entre les étages i et i-1 commence. Pour l'étage i on a une opération de lecture;donc C_i n'est pas actif. Par contre pour l'étage i-1,C_{i-1} est actif car on a une opération d'écriture. A la fin du comptage,c'est-à-dire fin de transfert d'informations entre les étages i et i-1,les signaux "fin de comptage" q_i et q_{i-1} passent de "1" à "0". Alors C_{0i}=1,Vc_{i-1} et Vc_i sont égaux à 0,v_{i-1}=0 et v_i=1. On est donc maintenant dans le cas où v_i=1.

Si v_i=1,1'autorisation de passage ap_{i+1} est activée (ap_{i+1}=1). Dans ce cas,on a un transfert d'informations entre les étages i+1 et i. Pour l'étage i+1,on a une opération de lecture et pour l'étage i une lecture. On revient dans le même cas précedent où le registre i-1 est vide. Mais dans ce cas,on remplace les i par i+1 et les i-1 par i.

Signal validation de comptage Vc_i: actifs à l'état haut

Supposons qu'on est sur la voie du haut:Vc_{0i} est actif si le signal d'écriture du registre R_{0i} est actif (C_{0i} passe de "1" à"0",c'est-à-dire on a un transfert des données entre les etages i+1 et i),ou si on a l'autorisation de passage correspondant (ap_{0i} =1,c'est-à-dire qu'on a un transfert des données entre les étages i et i-1). Vc_{0i} retourne à "0" sur le front descendant de q.

Pour réaliser ceci,on a besoin d'une impulsion de validation Iv active à l'état bas.Iv doit être synchrone avec l'horloge de comptage sinon le pre-

mier transfert de données est écourté. Iv est mis à "0" dès que C_{0_i}=0 ou ap_{0_i}=1 et retourne à "1" sur le front descendant de l'horloge.

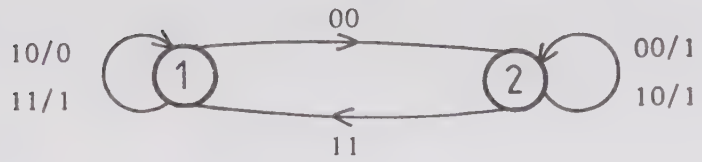
On a donc le graphe de fluence suivant:

où les entrées sont:-Iv:impulsion de validation

-q:signal fin de comptage (bit de plus fort poids du compteur de transfert de données)

où la sortie est:-Vc_{0_i}:signal de validation de comptage

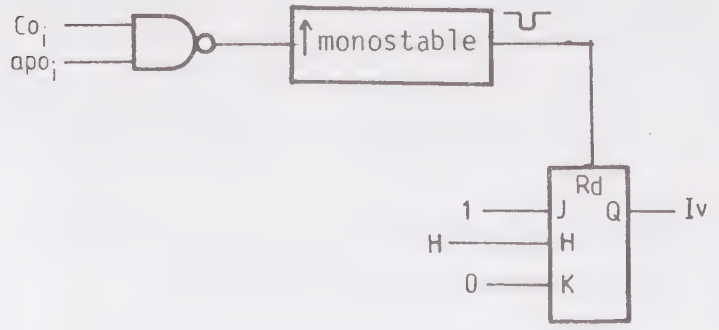
Iv q / Vc_{0_i}



Les équations logiques sont:

$y = \overline{q}\ Y + \overline{Iv}\ \overline{Y}$	pour la variable secondaire
$J = \overline{Iv}$	avec une bascule JK
$K = q$	
$Vc_{0_i} = \overline{q} \uparrow \overline{Y}$	pour la sortie

Iv doit être à"zéro" sur le front montant de ap_{0_i} ou sur le front descendant de C_{0_i} et retourne à "1" sur le front descendant de l'horloge. Elle est donc obtenue à partir d'une bascule JK , dont la borne de forçage à "0" Rd est reliée à un monostable qui délivre une impulsion au basculement de C_{0_i} ("1" à "0") ou de ap_{0_i} ("0" à "1"). Les entrées JK sont à "10" ce qui permet à chaque top d'horloge de mettre la sortie à "1".



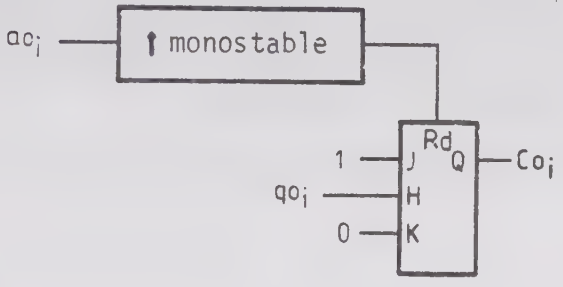
Signaux "autorisation d'écriture" des registres C_{0_i} et C_{1_i} : actifs à l'état bas

Soit l'étage i et la voie 0, l'autre voie étant identique à celle étudiée. C_{0_i} est actif quand on a un transfert d'information entre les étages i et i+1. Pour l'étage i on a un cycle d'écrite $C_{0_i} = 0$.

C_{0_i} passe de 1 à 0 sur le front montant de a_{0_i} et retourne à 1 sur le front descendant du signal fin de comptage q_{0_i}.

Ceci est réalisé par une bascule J K dont les entrées J et K sont à 1. La borne de forçage à 1 est aussi à 1. Sur le front montant de ap_{i+1}, un circuit délivre une impulsion négative sur l'entrée de forçage à 0 de la bascule.

A la fin du comptage (cycle d'écriture), q_{0_i} passe à 0 et fait basculer C_{0_i} de 0 à 1.



Réalisation des signaux de sélection des registres P,C,X,Y : actifs à l'état bas

Sp est actif quand le compteur est à 0)
Sc " " " " 1)
Sx " " " " 2) quand $Vc_i = 1$ (actif)
Sy " " " " 3)

ce qui donne

$$Sp = Vc_i \uparrow \overline{Q}_0 \uparrow \overline{Q}_1$$
$$Sc = Vc_i \uparrow Q_0 \uparrow \overline{Q}_1$$
$$Sx = Vc_i \uparrow \overline{Q}_0 \uparrow Q_1$$
$$Sy = R \uparrow Q_0 \uparrow Q_1$$

Réalisation des signaux d'écriture des registres P,C,X,Y : Les écritures se font sur le front montant du signal d'écriture du registre. Quand $C_{0_i} = 0$, c-à-d qu'on a écriture :

Cp = 0 quand le compteur est à 0 et remonte à 1 sur le front montant de l'horloge

Cc = 0 " " 1 " " "

"

Cx = 0 " " 2 " " "

"

Cy = 0 " " 3 " " "

"

Ce qui donne

$$\begin{aligned} Cp &= \overline{Q_0} \uparrow \overline{Q_1} \uparrow \overline{C_{0_i}} \uparrow \overline{H} \\ Cc &= Q_0 \uparrow \overline{Q_1} \uparrow \overline{C_{0_i}} \uparrow \overline{H} \\ Cx &= \overline{Q_0} \uparrow Q_1 \uparrow \overline{C_{0_i}} \uparrow \overline{H} \\ Cy &= Q_0 \uparrow Q_1 \uparrow \overline{C_{0_i}} \uparrow \overline{H} \end{aligned}$$

Signaux d'état des registres v_{0_i} et v_{1_i} : actifs à l'état haut

Un registre est vide si le transfert des informations entre ce registre et le registre destinataire de l'étage suivant est achevé, c'est-à-dire sur le front montant du signal "demande de passage" correspondant.

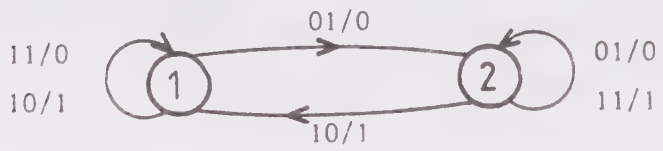
Pour les deux registres de l'étage, l'autre voie est identique à celle étudiée.

Soit l'étage i et la voie 0 : v_{0_i} = 1 (actif) sur le passage de dp_{0_i} de 0 à 1

v_{0_i} = 0 sur le passage de C_{0_i} de 0 à 1 (après écriture du registre R_{0_i}).

Le graphe de fluence correspondant est le suivant :

$dp_{0_i}\ C_{0_i}\ /\ v_{0_i}$



Les équations logiques sont : $y = \overline{dp_{0_i}}\ \overline{Y} + C_{0_i}\ Y$ pour la variable

$J = \overline{dp_{0_i}}$)

$K = \overline{C_{0_i}}$) avec une bascule J K

$v_{0_i} = Y\ dp_{0_i} + \overline{C_{0_i}}$)

Circuit d'état de la voie :

C'est un circuit combinatoire dont les sorties O_i sont fonction des entrées I_i, suivant l'état des signaux de destination et d'autorisation de passage.

On a le tableau suivant :

$d0_{i-1}$	$d1_{i+1}$	$ap0_{i+1}$	$ap1_{i+1}$	direction	O_0 O_1
X	X	0	0	- -	X X
X	1	0	1		X I_1
X	0	0	1		I_1 X
0	X	1	0		I_0 X
1	X	1	0		X I_0
0	1	1	1		I_0 I_1
1	0	1	1		I_1 I_0

pour les cas 00 11 et 11 11 où on a un conflit, le problème ne se pose plus car il est déjà traité au niveau du circuit d'élaboration des signaux autorisation de passage. On se ramène donc aux cas déjà cités ci-dessus.

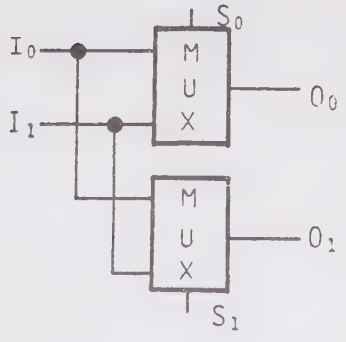
Pour réaliser ceci, on a deux multiplexeurs dont les entrées sont reliées en parallèle et dont les deux sorties représentent O_0, O_1.

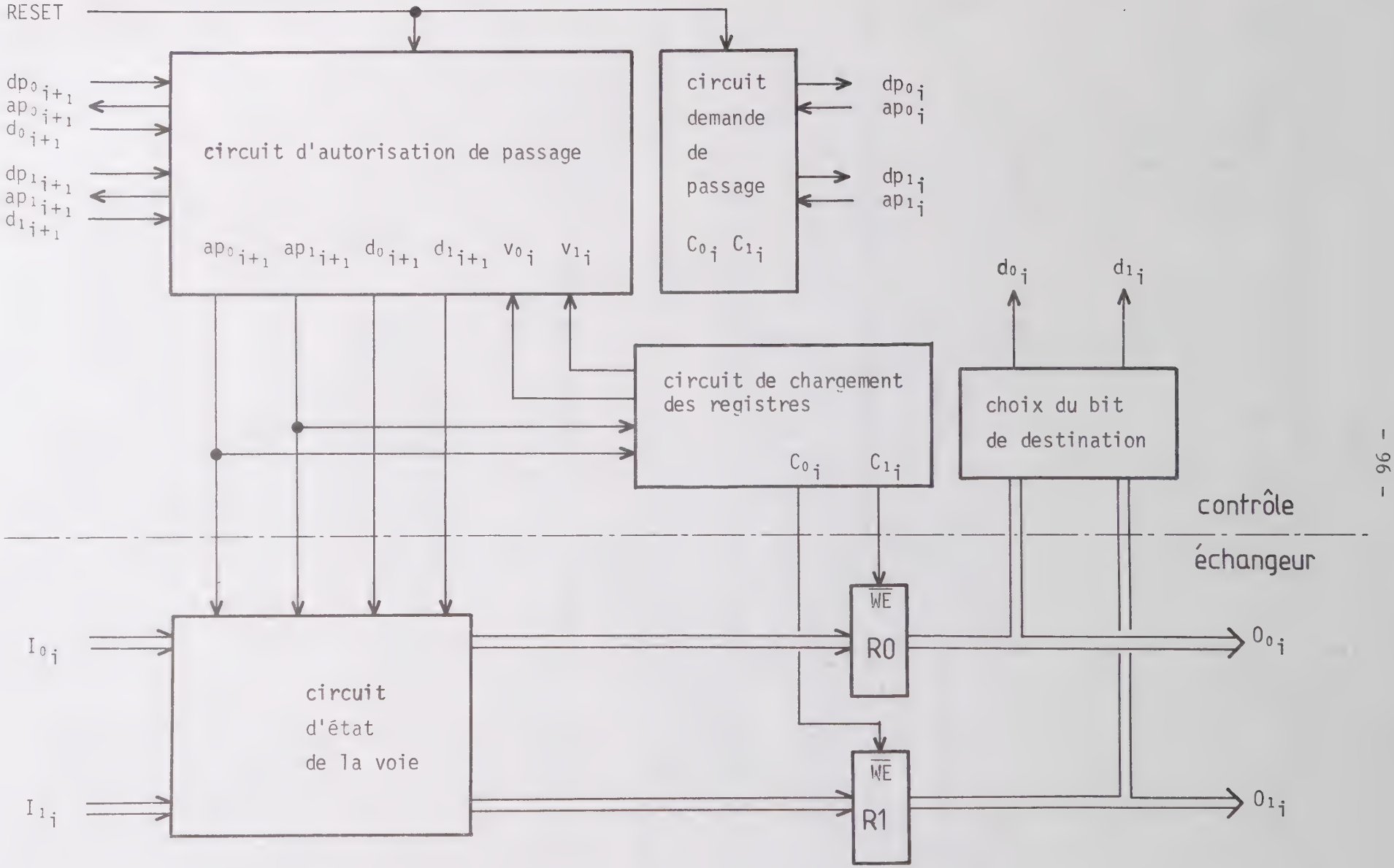
La fonction du multiplexeur est : Sj = 0 Oj = I_0

Sj = 1 Oj = I_1

Les équations logiques des deux sélections S_0 et S_1 sont :

$S_0 = \overline{d1_{i+1}}\ ap1_{i+1}$
$S_1 = d1_{i+1}\ ap1_{i+1}$





aiguilleur nºi

1.3.-Conception du circuit de décodage :

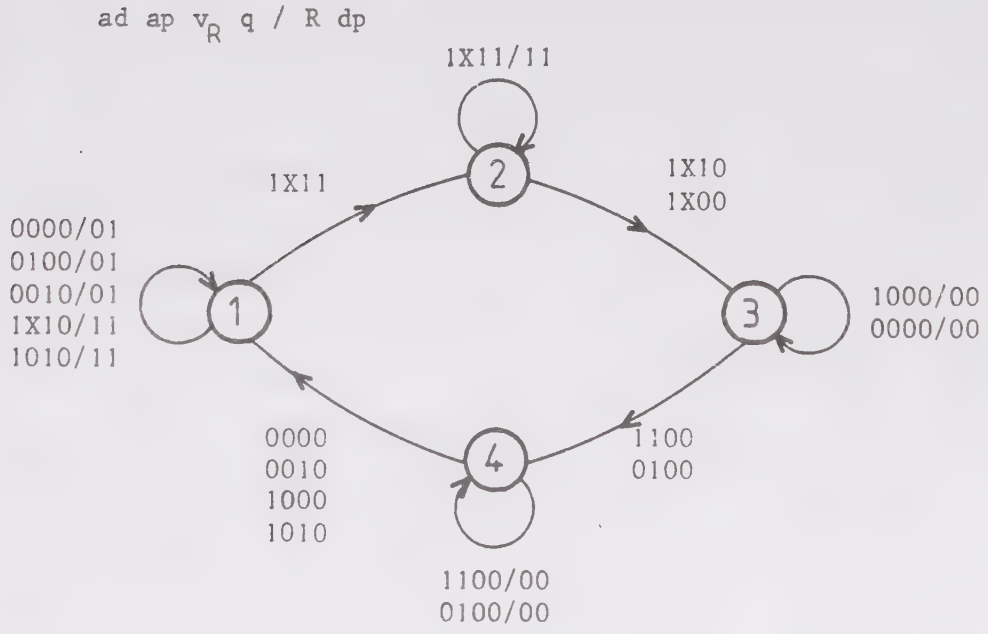
Il fait l'acquisition des données dans le tampon récepteur. Il teste l'état de celui-ci au moyen des signaux $\overline{WE}$ et "vide".

Si le tampon est vide, le circuit de décodage ne fait rien et attend que le tampon ne soit plus vide. Dès que l'information contenue dans son registre interne est passée (front descendant du signal "autorisation de passage" ou état du signal vide), si aucune écriture du tampon ne se fait et que le tampon n'est pas vide, alors le circuit active le signal de lecture R et écrit la donnée dans son registre interne. Si un cycle d'écriture du tampon est en cours, il attend la fin de celui-ci pour faire le cycle d'acquisition de données.

Une fois que la donnée est acquise, il libère le tampon en désactivant R. Dans ce cas, il fait le décodage de la destination et envoie la demande de passage ap qui reste active tant que la donnée n'est pas transmise dans le registre de destination correspondant (front descendant de l'autorisation de passage) et le cycle recommence.

R doit rester actif tant qu'on n'a pas fait le chargement du registre (front descendant de q).

Le graphe de fluence correspondant pour R et dp est le suivant :



$ad = \overline{\overline{WE}.\overline{v}}$ = autorisation de demande actif à l'état haut

ap = autorisation de passage

v_R = état du registre

q = fin de comptage

R = lecture du tampon

dp = demande de passage

v = signal vide du tampon

Les équations logiques sont : $\boxed{\begin{aligned} y_1 &= (\overline{ap}+\overline{Y}_2)Y_1+q\ \overline{Y}_1 \\ y_2 &= (ap+Y_1\overline{q})Y_2+Y_1\overline{q}\ \overline{Y}_2 \end{aligned}}$ pour les variables secondaires

ce qui donne pour des bascules J K

$$J_1 = q$$
$$K_1 = \overline{ap} \downarrow \overline{Y_2}$$
$$J_2 = \overline{q}\ Y_1 = q \downarrow \overline{Y_1}$$
$$K_2 = ap \downarrow (q \downarrow \overline{Y_1})$$

et $\boxed{\begin{aligned} R &= q+\overline{Y}_1\overline{Y}_2 ad+\overline{Y}_1 ad\ \overline{ap} \\ dp &= \overline{Y_1}\ \overline{ap}+\overline{Y}_1\overline{Y}_2+q \end{aligned}}$ pour les sorties

1.3.1.-Circuit d'écriture du registre :

Les données à charger sont : P,C,X,Y. Comme on a quatre données, il est préférable de les envoyer en quatre fois, cela augmente bien sûr le temps de chargement mais diminue la taille du bus de données quand celle-ci est assez importante. Le signal R établi dans le paragraphe précédent est composé avec l'horloge H pour former le signal composite RH. Ce signal attaque l'horloge du compteur lecture du tampon récepteur.

Comme le compteur lecture du tampon récepteur est bloqué à la dernière adresse présente qui n'a pas été lue, sur le front montant de R on fait une écriture du registre P. Pour cela on active le signal d'écriture Cp correspondant à ce registre. Au prochain top d'horloge, Cp est désactivé et Cc activé pour faire une écriture dans le registre C. Au deuxième top d'horloge, Cc est désactivé et Cx activé pour faire une écriture dans le registre X. Au troisième top, Cx est désactivé et Cs activé. Au quatrième top, Cy est désactivé en même temps que R, ce qui correspond à la fin de comptage. A ce moment, on a dans le compteur lecture du tampon récepteur l'adresse de la donnée P du prochain point.

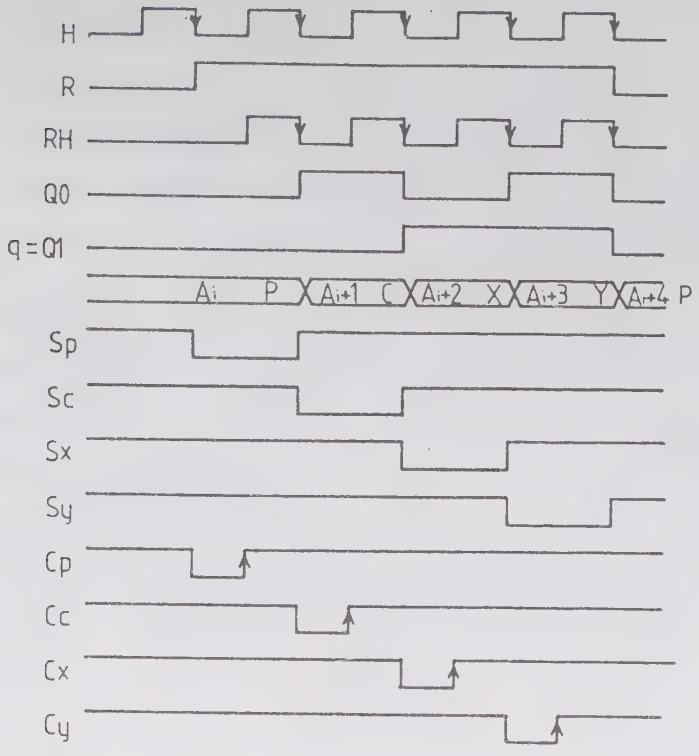
RH est aussi relié à un compteur par quatre, dont la sortie Q_1 est le signal fin de comptage q.

Dans le cas où les données passent en n fois, on utilise un compteur par n et q vaut Q_{n-1} si Q_{n-1} est la sortie du compteur de plus fort poids.

La période de l'horloge est réglée en fonction du temps de cycle de lecture des mémoires composant le tampon.

On fait toujours une sélection individuelle de chaque registre contenant chaque information point à chaque fois que l'on fait un cycle de lecture ou d'écriture.

Le principe est résumé par le diagramme de temps suivant :



Réalisation des signaux de sélection des registres P,C,X,Y : actifs à l'état bas

Sp est actif quand le compteur est à 0)
Sc " " " 1)
Sx " " " 2) et quand R est actif (R=1)
Sy " " " 3)

ce qui donne :

$$Sp = R \uparrow \overline{Q_0} \uparrow \overline{Q_1}$$
$$Sc = R \uparrow Q_0 \uparrow Q_1$$
$$Sx = R \uparrow \overline{Q_0} \uparrow Q_1$$
$$Sy = R \uparrow Q_0 \uparrow Q_1$$

Réalisation des signaux d'écriture des registres P,C,X,Y : comme les écritures se font sur le front montant du signal d'écriture du registre, C_i est à zéro, sur le front descendant de S_i, et remonte à 1 sur le front montant de H.

Ce qui donne :

$$Cp = \overline{Q_0} \uparrow \overline{Q_1} \uparrow R \uparrow \overline{H}$$
$$Cc = Q_0 \uparrow \overline{Q_1} \uparrow R \uparrow \overline{H}$$
$$Cx = \overline{Q_0} \uparrow Q_1 \uparrow R \uparrow \overline{H}$$
$$Cy = Q_0 \uparrow Q_1 \uparrow R \uparrow \overline{H}$$

Réalisation de l'information de destination : les coordonnées y_i sont transformées en mot de destination qui correspond au numéro de zone.

Soit z le nombre de zones, Dv la définition verticale ; donc pour :

$$y_i \in \left[0, \frac{Dv}{z} - 1\right] \quad , \quad D = 0$$

$$y_i \in \left[\frac{Dv}{z} , \frac{2Dv}{z} - 1\right] \quad , \quad D = 1$$

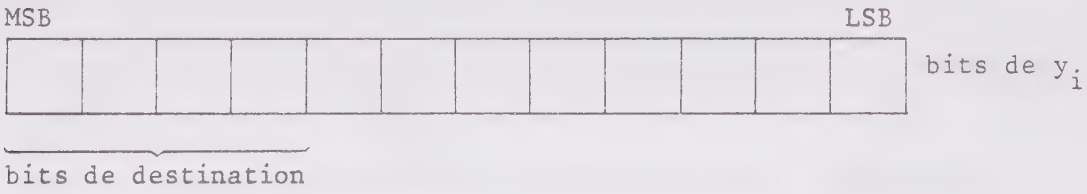
etc.

$$y_i \in \left[(z-1)\frac{Dv}{z} , Dv-1\right] \quad , \quad D = z-1$$

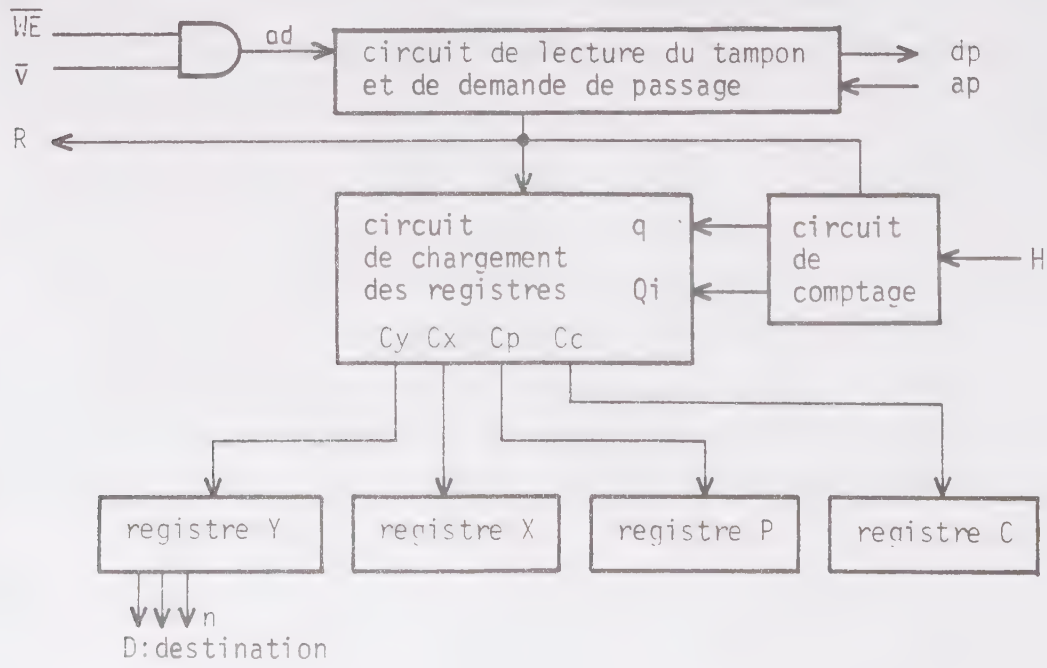
On voit, d'après le tableau ci-dessus, que $D = E\{\frac{Dv}{z}\}$ où E désigne la fonction partie entière, qui est obtenue par $n = \log_2 \frac{Dv}{z}$ décalages successifs.

On voit aussi que la destination est obtenue directement par les bits dont le poids faible est le bit numéro $n = \log_2 \frac{Dv}{z}$. Le bit de poids fort est le bit de poids fort de l'ordonnée y_i. C'est la solution adoptée.

Exemple : pour Dv = 512 et z = 4, y_i a 12 bits. Les bits de destination sont donc les bits 7 à 11 de y_i. Car $\log_2 \frac{512}{4} = 7$.



1.3.2.-Schéma bloc du circuit de décodage :

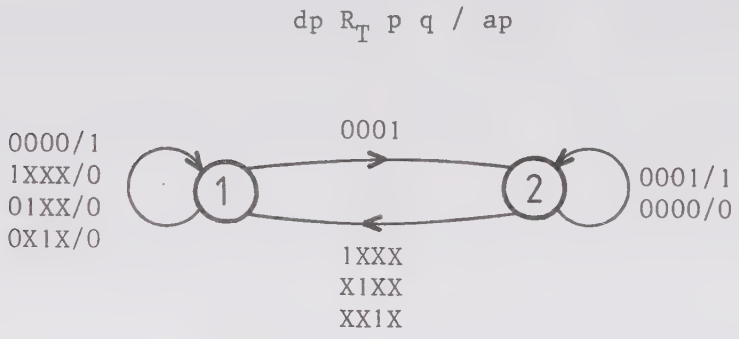


1.4.-Circuit d'écriture du tampon émetteur :

Il agit comme un demi-aiguilleur. Le dernier aiguilleur, dès que son registre interne est chargé, envoie une demande de passage à la zone correspondante et attend du circuit d'écriture une autorisation de passage. Cette demande de passage est maintenue tant qu'on n'a pas reçu l'autorisation de passage correspondante. L'autorisation de passage délivrée par le circuit d'écriture du tampon émetteur est obtenue à partir de l'état de ce tampon. L'autorisation de passage est validée si le tampon n'est pas plein et si aucun cycle de lecture n'est en cours.

Réalisation du signal autorisation de passage : il n'est actif que si on a une demande de passage et si le tampon n'est pas plein et si aucun cycle de lecture n'est en cours. Il reste actif tant qu'on n'a pas écrit les données dans le tampon. Il est désactivé sur le front descendant du signal fin de comptage q.

Le graphe de fluence est le suivant :



Les équations logiques sont

$$y = Y\,\overline{dp}\,\overline{R}_T\,\overline{p} + \overline{Y}\,\overline{dp}\,\overline{R}_T\,\overline{p}\,q$$

pour la variable secondaire

$$J = \overline{dp}\,\overline{R}_T\,\overline{p}\,q$$

$$K = \overline{dp} \uparrow \overline{R}_T \uparrow \overline{p}$$

avec une bascule JK

$$ap = \overline{Y}\,\overline{K} + J$$

pour la sortie

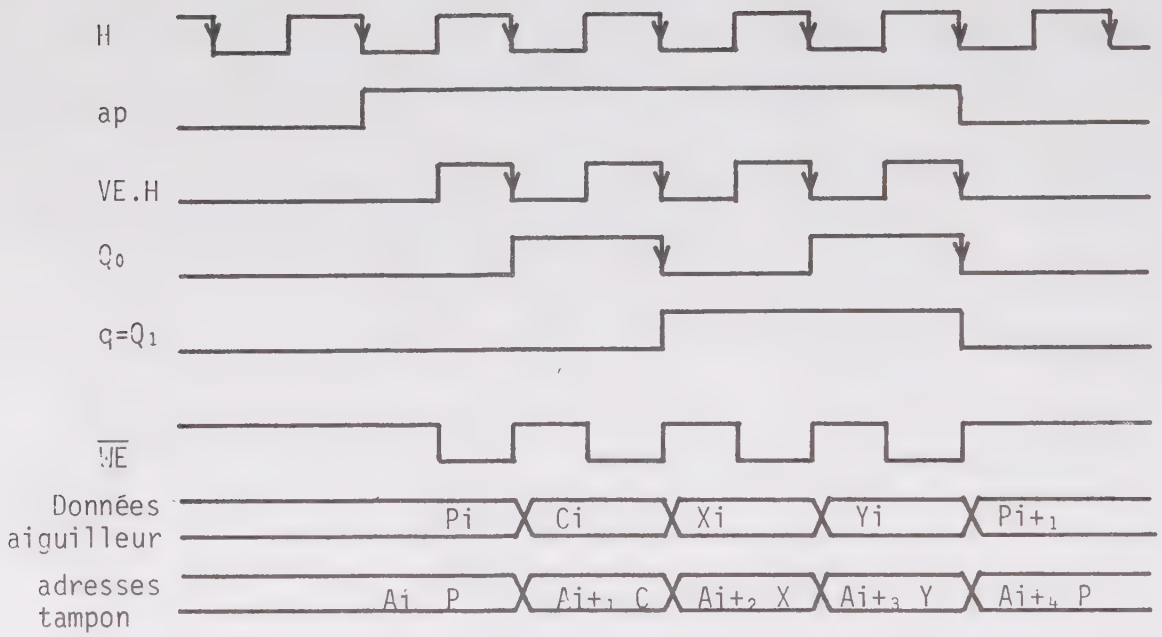
Réalisation du signal d'écriture $\overline{WE}$ du tampon : dès que le signal autorisation de passage est actif, il déclenche le signal d'écriture $\overline{WE}$ du tampon. En même temps, un signal validation d'écriture du tampon est activé et permet le comptage. Dès qu'on a fini l'écriture des données, front descendant de q, $\overline{WE}$ et VE sont désactivés.

Le signal validation d'écriture VE est le signal ap lui-même. Combiné avec H pour former le signal VE.H, il ne sert à incrémenter le compteur d'écriture ou de lecture que lorsque ap = 1 (c-à-d transfert des données du dernier aiguilleur

correspondant au tampon émetteur correspondant).

Le signal $\overline{WE}$ est le complément logique du signal composite VE.H. Le front descendant de $\overline{WE}$ écrit la donnée dans le tampon. Son front montant incrémente l'adresse du tampon et fait le passage d'une information à la suivante dans l'aiguilleur correspondant.

Le diagramme des temps est le suivant :



2.-Conception du circuit de mise à jour de l'image :

R(z), C(z) étant les rangées et colonnes des mémoires, on rappelle que :

$$\begin{cases} C(z) = \frac{256}{z} E\{\frac{X-\alpha(z)E\{\frac{X}{\alpha(z)}\}}{Dv}\} + E\{\frac{Y-\frac{Dv}{z}E\{\frac{zY}{Dv}\}}{y}\} + \beta(z)E\{\frac{Y-\gamma(z)E\{\frac{Y}{\gamma(z)}\}}{Dv}\} \\ R(z) = c(Y-yE\{\frac{Y}{y}\}) + E\{\frac{X-DvE\{\frac{X}{Dv}\}}{b}\} \end{cases}$$

en mode de travail "espace adressable" ;

et

$$C(z) = \frac{256}{z}(I-zE\{\frac{I}{z}\}) + E\{\frac{Y-\frac{Dv}{z}E\{\frac{zY}{Dv}\}}{y}\}$$

$$R = c(Y-yE\{\frac{Y}{y}\}) + E\{\frac{X}{b}\}$$

en mode de travail image :

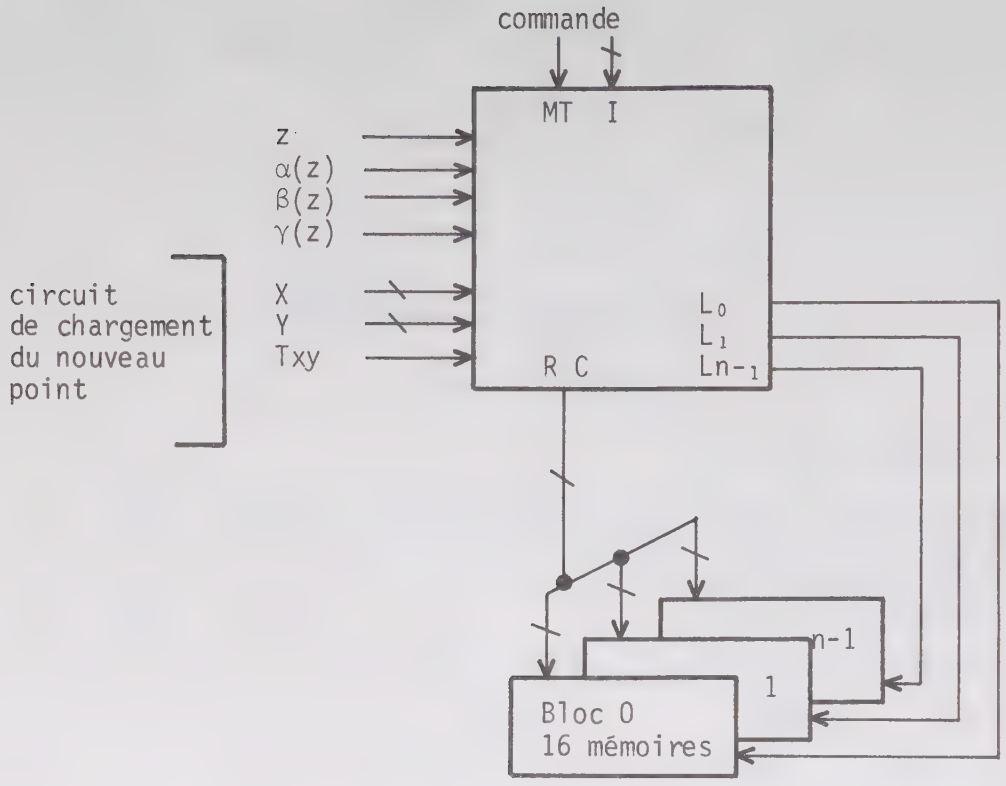
où b = nb de boîtiers devant être lus ensemble dans notre cas 16
c = nb de rangées par ligne " 64
Dv = définition verticale " 1024
y = nombre de lignes par colonne " 4
I = numéro de l'image
r = nb total de rangées d'un boîtier mémoire

z	1	2	4	8	16	32	64
$\alpha(z)$	Dv	2Dv	2Dv	4Dv	4Dv	8Dv	8Dv
$\beta(z)$	r	r	$\frac{r}{2}$	$\frac{r}{2}$	$\frac{r}{4}$	$\frac{r}{4}$	$\frac{r}{8}$
$\gamma(z)$	Dv	Dv	2Dv	2Dv	4Dv	4Dv	8Dv

2.1.- Circuit d'adressage : ce circuit permet d'adresser et de sélectionner les blocs de mémoires, suivant le signal mode de travail MT, suivant le numéro de l'image I [provenant de l'organe de commande en fonction du nombre de zones z de a(z), $\beta(z)$, $\gamma(z)$ réglées extérieurement] et suivant (X,Y) [(X,Y) provenant du circuit de chargement du nouveau point].

Quand le signal Txy [validant la transformation des coordonnées (X,Y) en adresses mémoire R et C et la sélection CS des blocs mémoires] est actif, R, C et CS sont activés et le restent jusqu'au prochain Txy, c'est-à-dire jusqu'au prochain chargement du nouveau point. Ce signal Txy assure aussi la sélection

individuelle des boîtiers, en fonction de X et Y, dans chaque lot.



Mode de travail dans un espace axb MT = 0

I = ϕ ϕ veut dire indifférent

Conditions de fonctionnement : a et b sont des puissances de 2. Soit à définir un espace adressable de 8192 x 8192 points. Physiquement, ceci correspond à 64 lots de 16 mémoires 64K. Un bloc de 16 mémoires 64K correspond à 1024 x 1024 points.

Le nombre de zones doit être une puissance de 2 sinon on n'aurait pas de zones indépendantes. $z \in [1,64]$

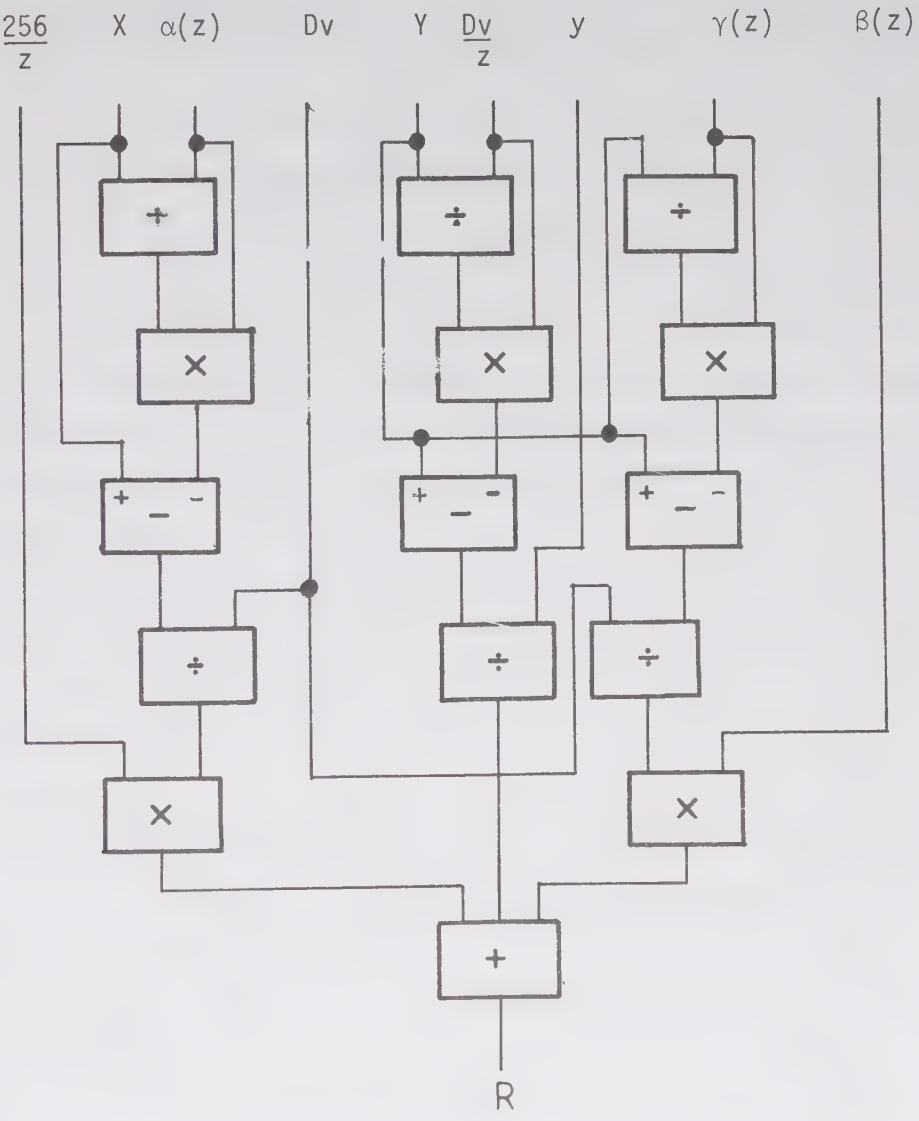
Nota : Il n'est pas nécessaire d'avoir les 64 blocs de 16 mémoires. Dans ce cas, seuls sont chargés les points intérieurs à l'espace défini par les blocs de 16 mémoires.

Mode de travail en image : MT = 1

I = numéro de l'image à mettre à jour

Conditions de fonctionnement : elles sont les mêmes que pour le mode de travail en espace adressable sauf que : $a_{max} = 1024$, $b_{max} = 1024$ et $I \in [0,63]$.

2.1.1.-Choix des opérateurs pour la réalisation de R,C,B : d'après les expressions de R,C,B, les opérations à faire sont des additions, des soustractions, des multiplications et des divisions entières, comme le montre la figure dessinée pour la variable R.



Les paramètres Dv, $\frac{256}{z}$, $\alpha(z)$, $\frac{Dv}{z}$, $\gamma(z)$ et $\beta(z)$ sont définis dès que l'on a choisi le nombre de zones.

Les variables sont donc X, Y et I quand on travaille dans le mode image.

Il existe plusieurs méthodes pour la réalisation de R,C et B :

- programme à l'aide de microprocesseur
- utilisation d'unité arithmétique flottante (type AMD 95II et TRW)
- utilisation d'opérateurs câblés (multiplieur, additionneur,...)
- utilisation de PROM et d'opérateurs câblés.

La solution programmée est à éviter, car elle est lente et nécessite l'emploi d'un microprocesseur pour chaque zone.

L'utilisation d'unité arithmétique n'est pas non plus avantageuse, car non seulement elle est coûteuse, mais en plus on n'utilise pas toutes ses performances.

Les seules solutions à retenir sont donc l'utilisation d'opérateurs câblés, et l'utilisation de PROM associés aux opérateurs câblés.

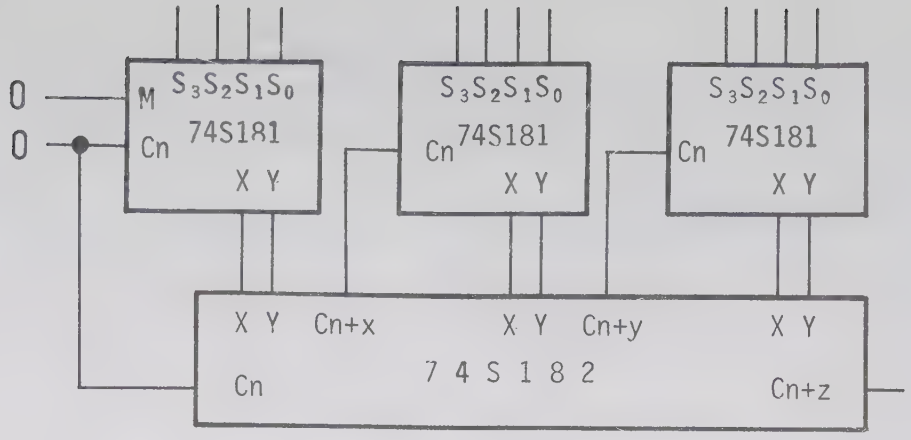
Etude de la solution câblée : si le bit de poids le plus fort du dividende est à gauche, la division entière est obtenue en faisant un décalage de n = $\log_2$ (diviseur) fois à droite du dividende. Actuellement, du fait de l'évolution technologique, on arrive à une fréquence de décalage de l'ordre de 80 MHz.

Le tableau ci-dessous donne le nombre de décalages à faire et le temps d'exécution d'une division entière en fonction du nombre de zones z.

z	1	2	4	8	16	32	64
nb de décalage	10	11	11	12	12	13	13
t_{div} ns	125	137,5	137,5	150	150	162,5	162,5

La multiplication est obtenue par décalages de sens contraire à celui de la division, ou par des multiplieurs.

L'addition et la soustraction binaire sont réalisées à l'aide de circuits à fonctions multiples rapides (type 82S82 de RTC ou 74S181) combinés avec des générateurs de retenue anticipée (type 74S182) pour augmenter la vitesse d'exécution des opérations à faire. Le temps d'exécution est de l'ordre de 34 ns, pour un nombre de bits compris entre 9 et 16.



Etude de l'utilisation des mémoires mortes associées à des opérateurs câblés :

L'idée est d'utiliser les mémoires mortes comme générateurs de fonction. La question qui se pose est : pourquoi ne pas utiliser uniquement les mémoires mortes pour générer les fonctions $R_z(X,Y)$, $C_z(X,Y)$ et $L_z(X,Y)$?
Réponse : comme ce sont des fonctions à deux variables (X,Y), il faudrait XxY cases mémoires pour générer la fonction ; dans notre cas il faudrait une taille mémoire 8192 x 8192 = 64 M bits !!!

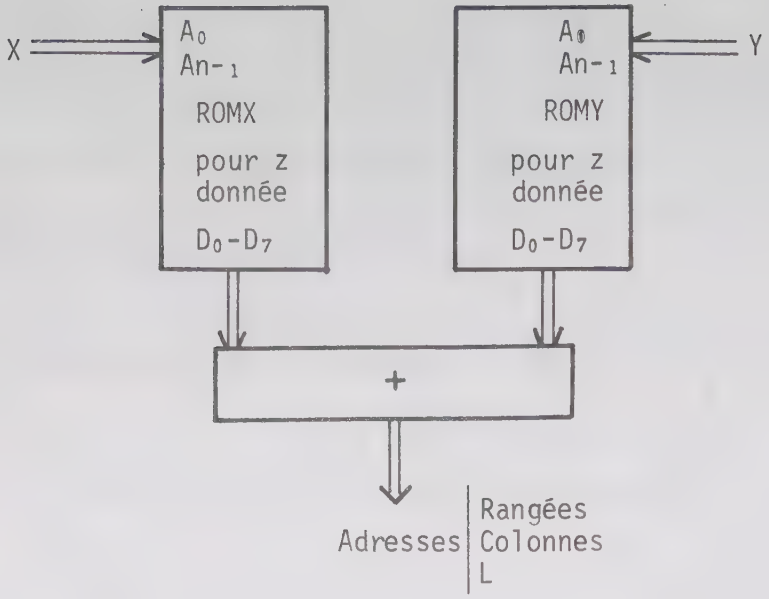
Les fonctions à réaliser sont du type $f_z(X,Y) = g_z(X) + h_z(Y)$
où $g_z(X)$ et $h_z(Y)$ sont de la forme $m\ E\{\frac{X \text{ ou } Y - n\ E\{\frac{X \text{ ou } Y}{n}\}}{p}\}$

La solution consiste donc à générer, par des mémoires mortes différentes notées ROMX ou ROMY, les fonctions $g_z(X)$ et $h_z(Y)$ pour z donné et à additionner celles-ci pour avoir la fonction $f_z(X,Y)$. On a donc, pour un nombre de zones z donné, une ROMX et une ROMY correspondantes.

Le problème revient donc à un décodage par mémoires mortes, où X et Y sont les adresses de chaque ROMX et ROMY. Les données sont les valeurs des fonctions $g_z(X)$ et $h_z(Y)$ et sont additionnées avec le type d'additionneur décrit précédemment.

Avec cette méthode, on gagne une rapidité d'exécution avec un coût faible. Le temps de calcul des adresses est réduit au temps d'accès des ROM, ajouté au

temps de calcul de l'additionneur.



Il faut 8 bits de données car $256 = 2^8$

Réalisation des ROMX et ROMY : pour la ROMX ou la ROMY donnée, on a :

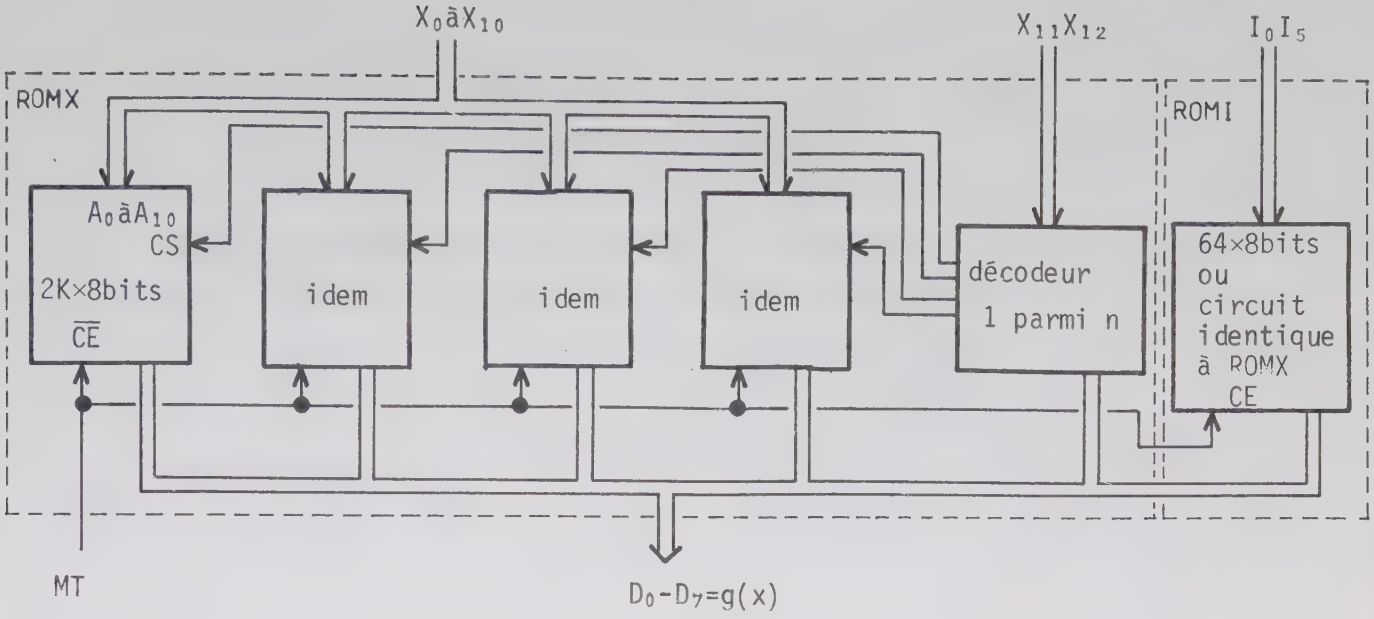
X ou Y = 0	Ax ou Ay = 0
X ou Y = 1	Ax ou Ay = 1
....	....
X ou Y = 8191	Ax ou Ay = 8191

Pour (X,Y) $\in$ (8192,8192), les adresses Ax et Ay ont 13 bits chacune.

Les mémoires mortes TTL utilisées existant actuellement sont de type 2k x 8 bits (82S191 de Signetics, 63S1681 de Monolithic (65ns), MCM 76161 de Motorola). Elles ont un temps d'accès maximum de 100 ns.

Pour réaliser la ROMX ou la ROMY, il faut 4 boîtiers pour X. De même pour Y. Les adresses X_0 à X_{10} sont en parallèle. X_{11} et X_{12} vont dans un décodeur "1 parmi n" pour sélectionner les boîtiers.

Suivant le mode de travail désiré (MT = 0 ou 1), on choisit l'un ou l'autre des deux blocs mémoires ROM présentés ci-après pour la réalisation de l'adresse colonne. Pour l'adresse rangées, les ROMX, Y sont les mêmes quelque soit le mode de travail.

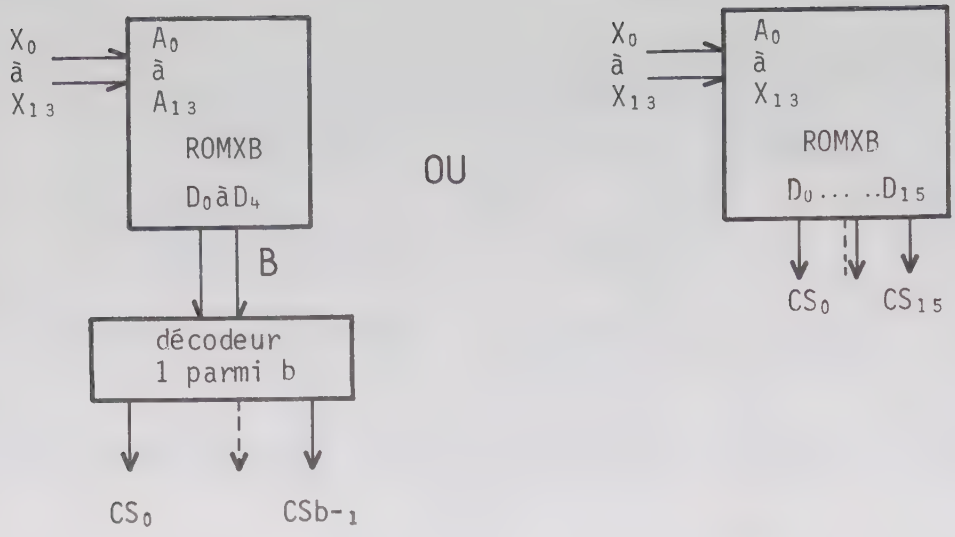


pour la ROMY, au lieu d'avoir X_0 à X_{12}, on a Y_0 à Y_{12} et on n'a plus besoin de ROMI.

Elaboration du rang de boîtiers B : on a $B = X - b\ E\{\frac{X}{b}\}$

Le principe est aussi le même. On a une ROMXB pour la donnée (b étant fonction de la vitesse des mémoires utilisées). b est constante pour une définition horizontale donnée. Deux possibilités existent : soit sélectionner directement le boîtier dans lequel se trouve X. Dans ce cas on a une ROM 2k x b bits. Ceci n'est avantageux que lorsque b est faible;

soit sélectionner le boîtier dans lequel se trouve X par l'intermédiaire d'un décodeur 1 parmi b. Ce décodeur est relié sur le bus de données de la ROMXR.



Elaboration des adresses mémoires :

Les adresses rangées et colonnes, provenant du circuit d'adressage et du circuit de visualisation et de rafraîchissement des mémoires, sont multiplexées pour former les adresses rangées et colonnes des mémoires.

En période de lecture-écriture (AEL = 0, RFSH = ϕ), suivant R/C, on a sur le bus d'adresse soit R_{EL} soit C_{EL}.

Pendant la période de visualisation (AEL = 1, RFSH = 1), suivant R/C, on a sur le bus d'adresse soit Rv soit Cv.

Pendant la période de rafraîchissement (AEL = 1, RFSH = 0), quel que soit R/C, on a toujours sur le bus d'adresse R_{RF}.

On a donc :

$$A = \overline{AEL}.R/C\ R_{EL} + \overline{AEL}.\overline{R/C}.C_{EL} + AEL.\overline{RFSH}.R_{RF} + AEL.R/C.RFSH.Rv + AEL.\overline{R/C}.RFSH.Cv$$

Sélection des lots :

Mode de travail image : dans un lot de z zones, on peut mettre z images. Dans le domaine défini par les conditions de fonctionnement, on a z = 64. Donc I_{max} dans un lot = 63, I étant le numéro de l'image. Les images

0 à z-1 sont contenues dans le lot 0

z à 2z-1 " " " 1

etc.

on a : $L = E\{\frac{I}{z}\}$

La fonction donnant le numéro des lots est aussi réalisée par un ROM. La sélection des lots est obtenue soit directement, soit par l'intermédiaire d'un décodeur 1 parmi n, comme dans le cas de la sélection des boîtiers.

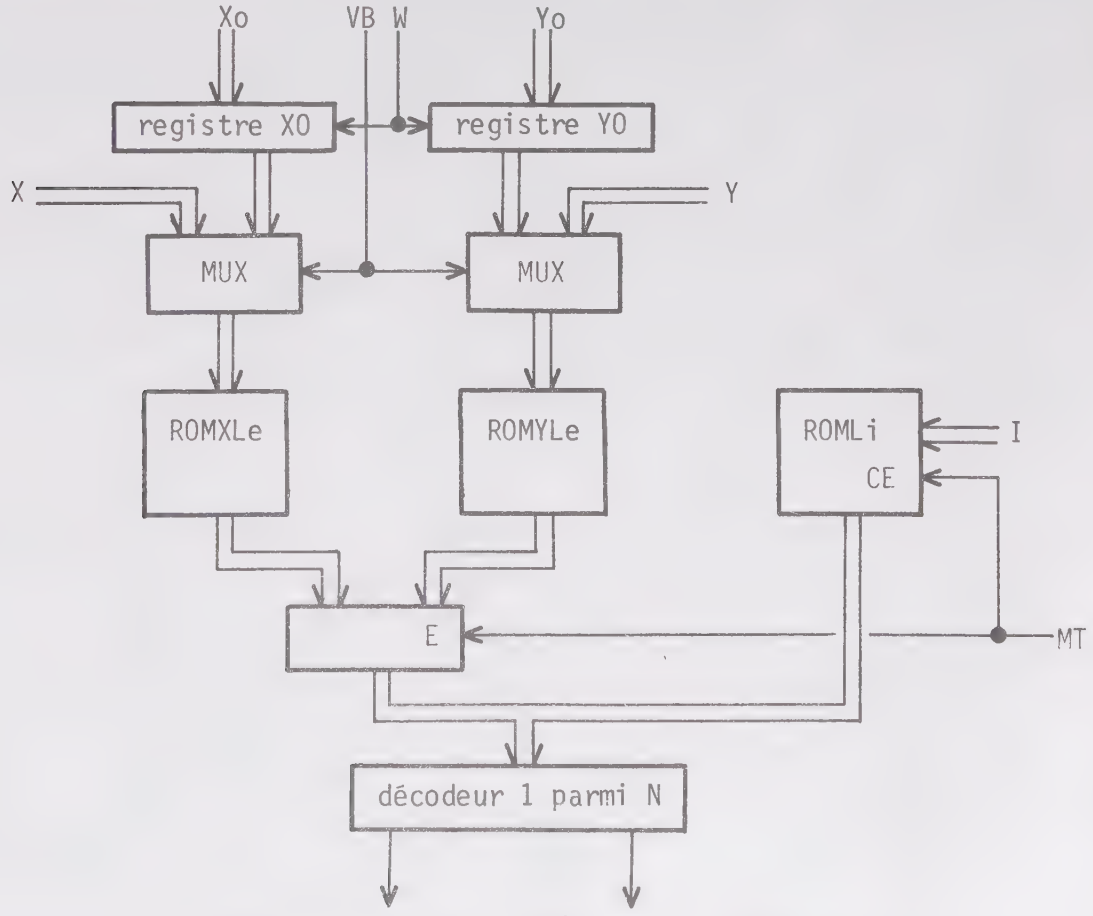
Mode de travail espace : le principe est le même que pour l'obtention des adresses rangées R et colonnes C. En adresses, on a X pour la ROMX et Y pour la ROMY. Les données des deux ROM sont additionnées pour former Le.

Que ce soit en mode image ou en mode espace, la sélection des boîtiers se fait par les signaux RAS et CAS. La réunion des boîtiers pour constituer un lot est faite manuellement.

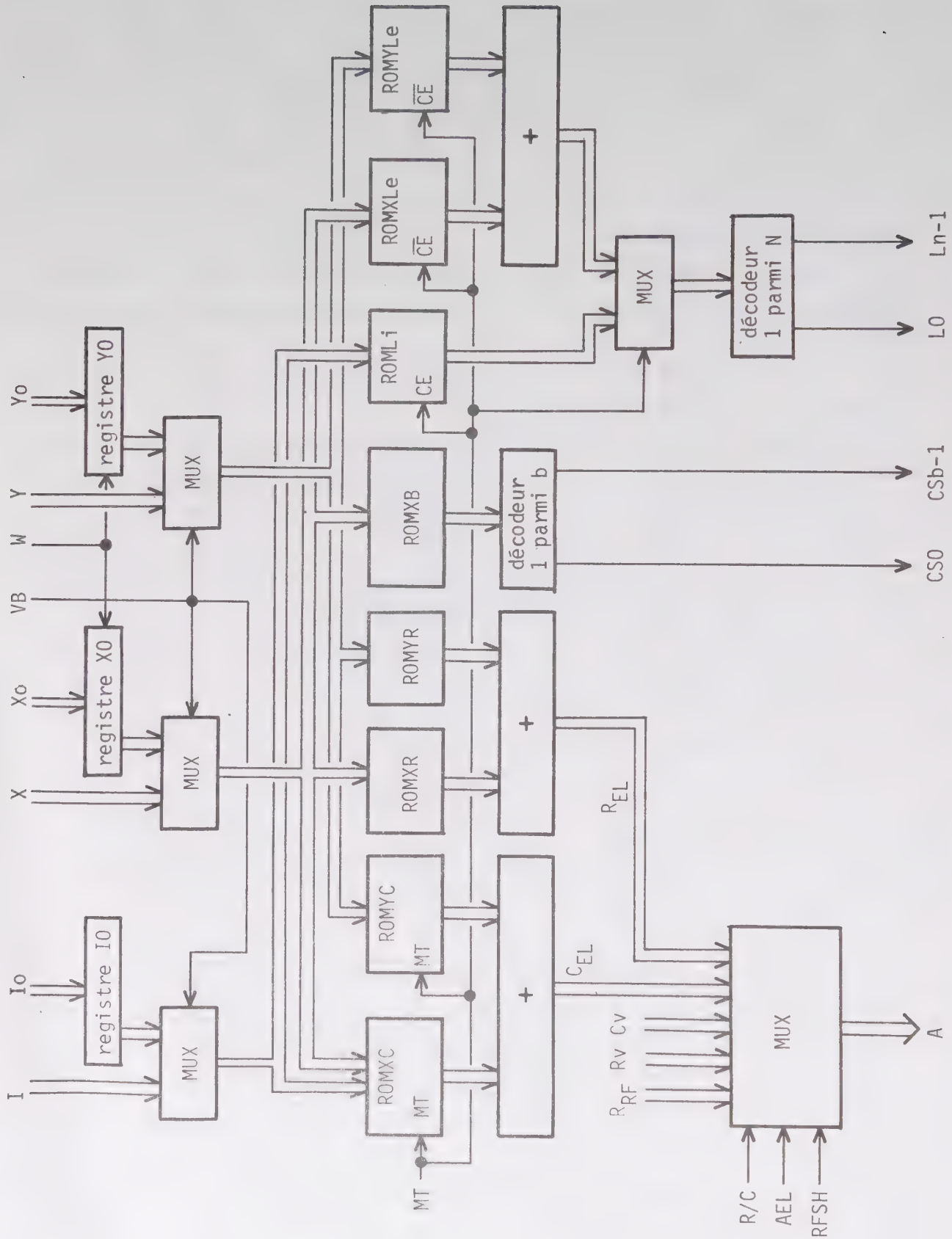
Pendant la période de visualisation (VB=1), les adresses de ROMXLe et ROMYLe sont respectivement X_0 et Y_0 (adresses coordonnées de l'origine de l'espace à visualiser).

Les coordonnées X_0, Y_0 sont mémorisées dans deux registres X_0 et Y_0 et peuvent être modifiées à tout moment.

En dehors de la période de visualisation (VB=0), ce sont les adresses (X, Y) du point courant qui sont les adresses des ROMXLe et ROMYLe.



Il en est de même pour les ROM rangées et colonnes.


Yo
registre YO
MUX
Y
W
VB
Xo
registre XO
MUX
X
Io
registre IO
MUX
I
ROMXC
MT
ROMYC
MT
ROMXR
ROMYR
ROMXB
ROMLi
CE
ROMXLe
CE
ROMYLe
CE
+
+
+
+
MUX
décodeur
1 parmi b
décodeur
1 parmi N
CSO
CSb-1
LO
Ln-1
MT
C_{EL}
R_{EL}
R_{RF}
Rv
Cv
MUX
A
R/C
AEL
RFSH

2.1.2.-Elaboration de RAS_M et CAS_M : RAS_M et CAS_M sont réalisés à partir de CS, L de RAS et CAS provenant d'autres circuits.

Un lot correspond à z blocs de 16 mémoires. Au niveau du boîtier mémoire i correspondant à un lot j, on doit avoir RAS_{Mi} = RAS et CAS_{Mi} = CAS.
Quand CS_i et L_j sont valides, c-à-d pour chaque (X,Y) car $CS_i = f(X)$ et $L_j = g(X,Y)$.
Comme les RAS et CAS sont actifs à l'état bas, quand CS_i et L_j ne sont pas valides ensemble, alors RAS_{Mi} et CAS_{Mi} doivent être au niveau haut. L_j et CS_i sont valides à l'état bas, ceci pour le mode lecture-écriture : c'est la sélection individuelle des boîtiers. Pendant la période de rafraîchissement des mémoires, tous les boîtiers contenus dans la zone sont sélectionnés avec RAS actif mais CAS au niveau 1.

Pendant la période de visualisation, tous les boîtiers sont aussi sélectionnés. RAS et CAS sont actifs pour le lot choisi. Pour les autres RAS_{Mi} = RAS et CAS_{Mi} = 1, ce qui permet de rafraîchir les mémoires du lot non visualisé quand on est en période de visualisation.

On voit donc que RAS_{Mi} est toujours actif, sauf pendant la période de lecture-écriture où, quand le boîtier n'est pas sélectionné, il vaut 1. C'est-à-dire RAS_{Mi} = RAS quand AEL = 1 ∀ CSi (période de visualisation ou de rafraîchissement)

= RAS quand AEL = 0 et CSi = 0

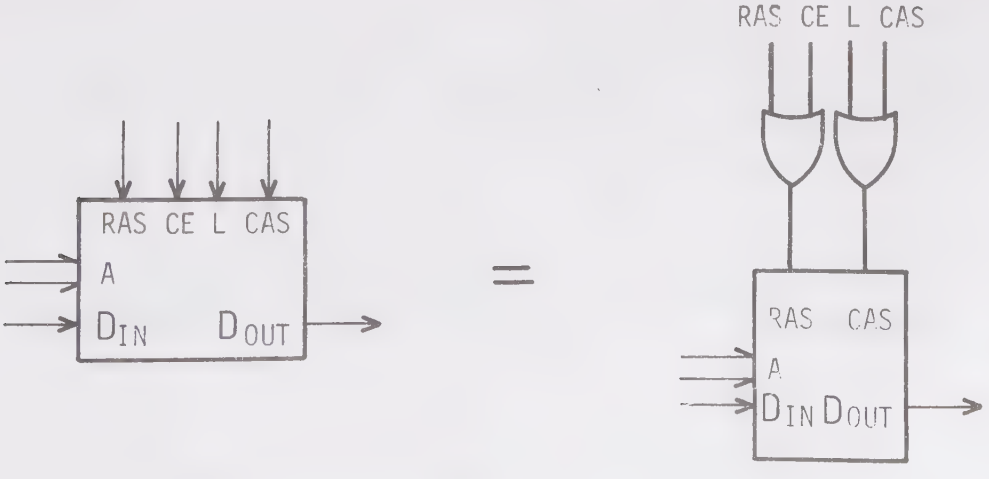
= 1 quand AEL = 0 et CSi = 1

Et CAS_{Mi} n'est égal à CAS que si le boîtier est sélectionné pendant la période d'écriture lecture, ou que si le lot correspondant est en période de visualisation. Dans les autres cas, il vaut 1.

Ce qui donne comme équations logiques de RAS_{Mi} et CAS_{Mi} :

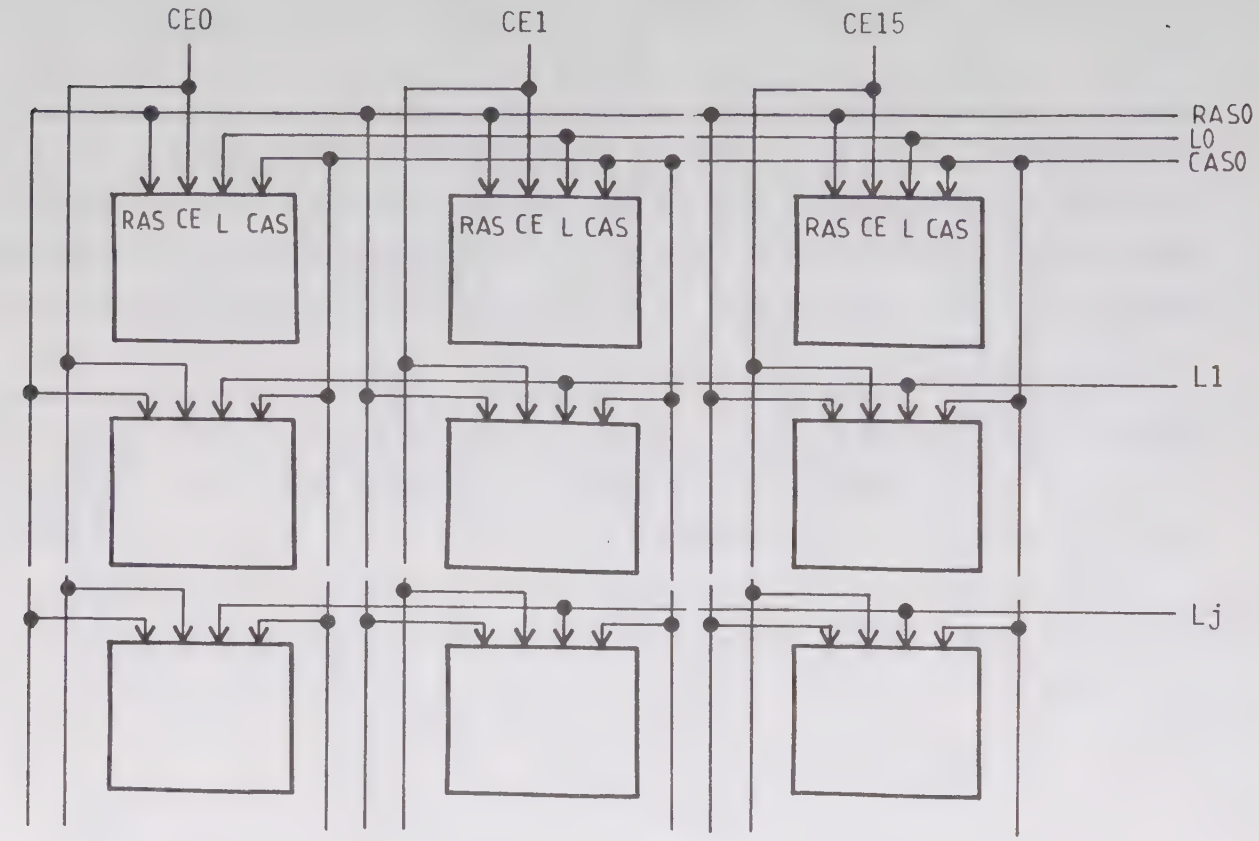
$$RAS_{Mi} = RAS + \overline{AEL}.CSi$$
$$CAS_{Mi} = CAS + L_j$$

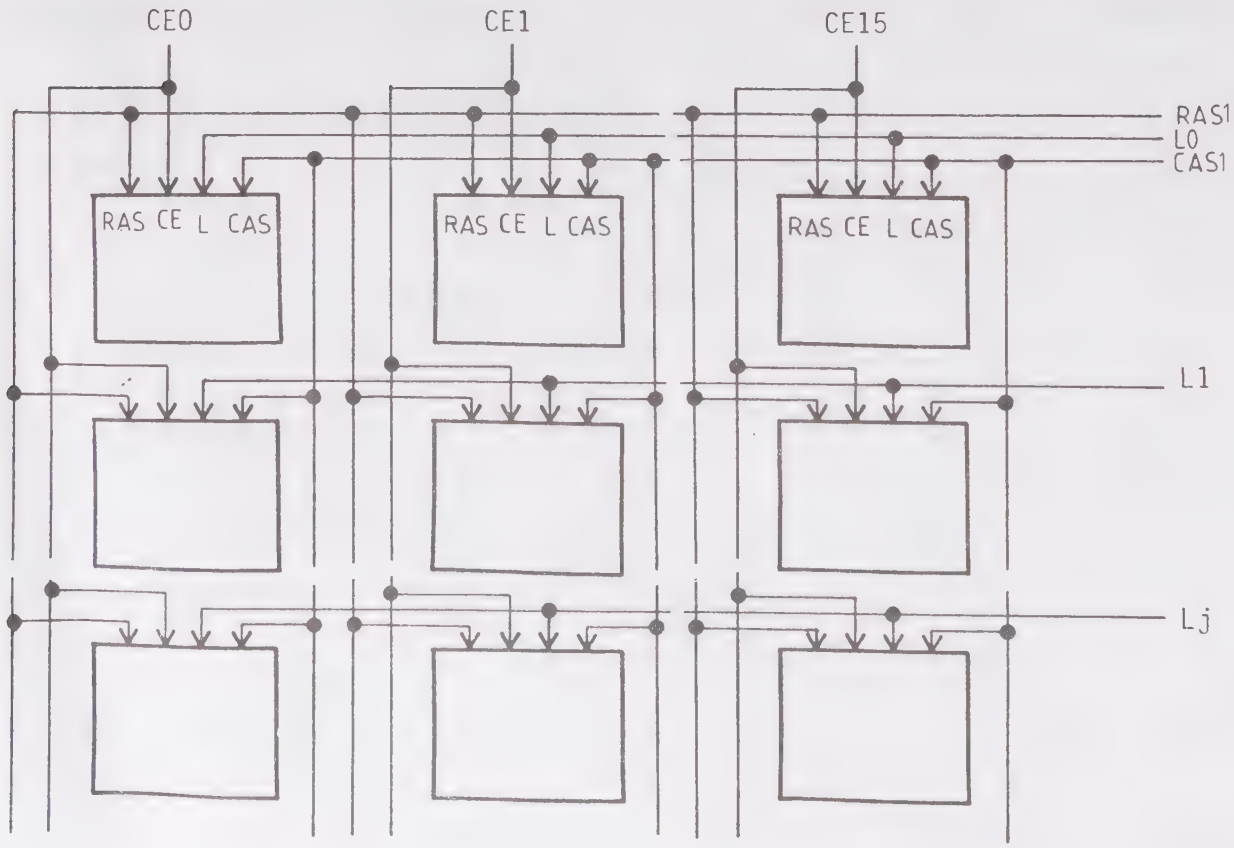
Par la suite, un boîtier mémoire est considéré comme suit :



CE peut être soit $\overline{AEL}$.CSi, soit CSi si toutes les sorties CS du décodeur parmi b de la sélection de boîtier sont au niveau 0 quand AEL = 1.

2.1.3.-Organisation des mémoires :



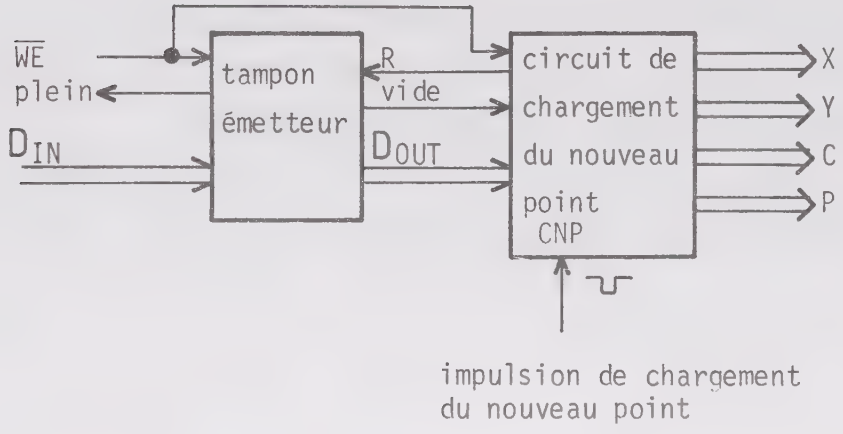


2.2.-Circuit de chargement du nouveau point :

Dès que le circuit reçoit l'ordre de chargement du nouveau point (impulsion de chargement du nouveau point), il fait l'acquisition des données provenant du tampon émetteur. Dans ce cas, il teste l'état du tampon au moyen des signaux $\overline{WE}$ et "vide". Si le tampon est vide ou s'il est en période d'écriture, le circuit ne fait rien.

Quand le tampon n'est ni vide, ni en période d'écriture, R est activé en même temps que le signal d'écriture des registres X,Y,C,P du nouveau point. Puis le signal de validation de comptage, dont le principe est identique pour tous les circuits d'écriture ou de lecture des données, est activé à son tour. Sur le front descendant du signal fin de comptage q_i, le comptage s'arrête, R et le signal d'écriture des registres ne sont plus activés, et on attend un autre ordre de chargement pour recommencer le phénomène.

Tant qu'on n'a pas le nouvel ordre de chargement, on a constamment les données valides à la sortie.



Réalisation du signal de lecture du tampon émetteur R :

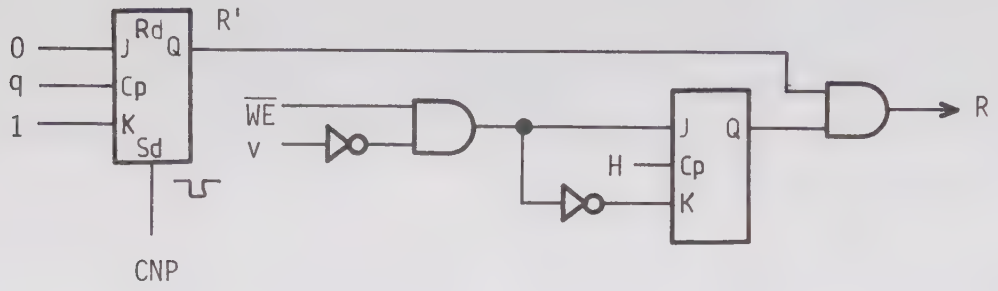
L'impulsion de chargement du nouveau point est mémorisée pour donner le signal R' qui, combiné avec $\overline{WE}$ et v, donne le signal R.

R' est obtenu à partir d'une bascule JK dont les entrées J et K sont à 01. L'apparition de l'impulsion CNP (reliée à la borne de forçage à 1) met à 1 R'.

L'horloge est reliée au signal fin de comptage q, dont le front descendant fait basculer la sortie R' de 1 à 0.

Quand on a reçu l'ordre de chargement et que le tampon n'est ni vide, ni en période d'écriture (c-à-d R' = 1, $\overline{WE}$ = 1 et v = 0), on doit avoir R = 1 et sur le front descendant de q, R doit passer à 0. $\overline{WE}\ \overline{v}$ est synchronisé avec l'horloge. Ceci est obtenu avec une bascule JK, dont J = $\overline{WE}\ \overline{v}$ et K = $\overline{WE}\ \overline{v}$.
On a donc R = Y R' car R' passe à 0 quand q passe à 0.

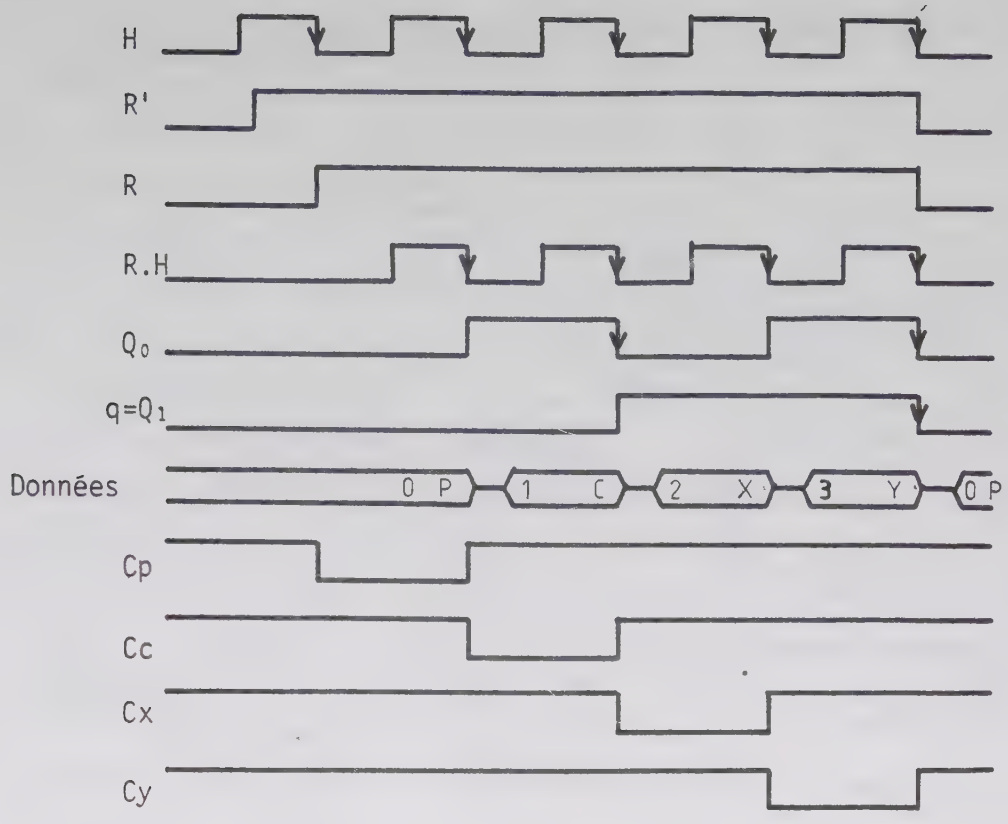
R ne passe à 1 que lorsque Y passe à 1 sur le front descendant de l'horloge.



R est le signal de lecture du tampon si les données passent en une seule fois. Dans le cas où les données passent en plusieurs fois, R sert à indiquer au tampon émetteur si on est ou non en période de lecture.

R composé avec H est le signal d'horloge du compteur de chargement.

Diaphragme des temps : on suppose que les données passent en quatre fois dans l'ordre suivant : P,C,X,Y.



Réalisation des signaux d'écriture des registres P,C,X,Y : les signaux sont actifs à l'état bas

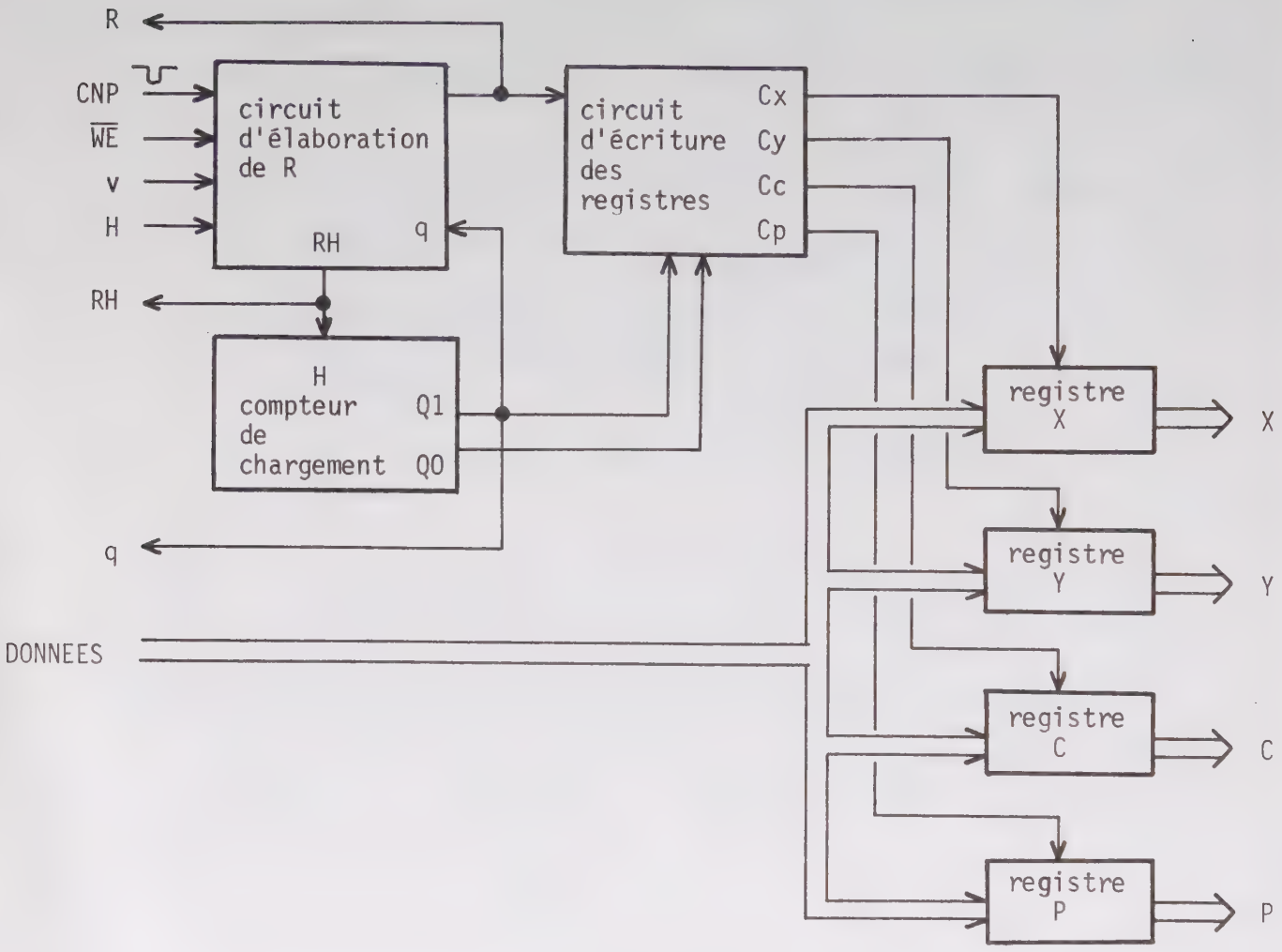
Cp est actif quand le compteur est à 0
Cc " " " 1
Cx " " " 2
Cy " " " 3
) et quand R est actif (R=1)

On a donc :

$$Cp = R \uparrow \overline{Q}_0 \uparrow \overline{Q}_1$$
$$Cc = R \uparrow Q_0 \uparrow \overline{Q}_1$$
$$Cx = R \uparrow \overline{Q}_0 \uparrow Q_1$$
$$Cy = R \uparrow Q_0 \uparrow Q_1$$

On écrit seulement dans le registre où Ci est actif.

Organisation du circuit de chargement du nouveau point :



2.3.-Circuit d'état de la zone :

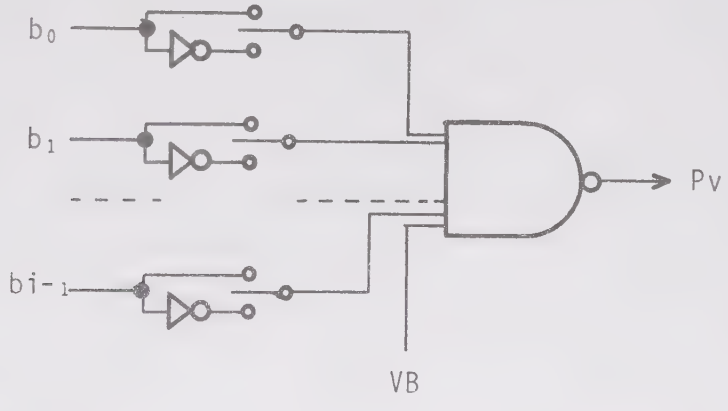
Il sert à indiquer si on peut écrire ou lire dans la mémoire d'image. Pour cela, il fait un décodage en permanence du numéro de zone en train d'être visualisée et, au moyen du signal de rafraîchissement de la zone RFSHi, il donne l'autorisation de lecture-écriture avec le signal autorisation lecture écriture AEL (actif = l'état bas). Ce signal est activé si la zone n'est ni en période de visualisation, ni en période de rafraîchissement ; sinon il n'est pas activé.

Décodage de la zone visualisée : le décodage doit permettre à la zone d'être placée dans n'importe quelle position.

Soit bi les bits du numéro de zone en train d'être visualisée et Pv la sortie du décodeur.

Si bi est égal au numéro de la zone en train être visualisée, alors Pv = 0 ; sinon Pv = 1. On utilise le même principe que pour faire une sélection par décodage complet. bi ou $\overline{bi}$ sont donc les entrées d'une porte NONET à i entrées, dont la sortie est à 0 si tous les i entrées sont à 1.

Pour permettre à la zone d'être placée dans n'importe quelle position, la porte NONET dispose de 2i entrées qui comportent i inverseurs.



Les commutateurs relatifs à chaque bit sont reliés à bi ou à $\overline{bi}$ suivant la position de la zone, de façon à ce que toutes les entrées soient à 1 quand la zone est visualisée.

Réalisation du signal autorisation de lecture-écriture : quand p_V = 0 et HB = 1 ou RFSH = 0, alors AEL = 1 : pas de lecture-écriture.

P_V = RFSH = 1 alors AEL = 0 : autorisation de lecture-écriture

ou P_V = 0 RFSH = 1 et HB = 0

Ce qui donne :

$$\overline{AEL} = RFSH\ (PV + \overline{HB})$$
$$AEL = RFSH \uparrow (\overline{PV} \uparrow HB)$$

2.4.-Circuit de lecture et de chargement de l'ancien point :

Dès la fin du chargement du nouveau point (front descendant du signal fin de comptage q du circuit de chargement du nouveau point), un ordre de chargement de l'ancien point est généré. A la réception de cet ordre, le circuit fournit les signaux RAS et CAS et le signal R/C pour le multiplixage des adresses.

Réalisation de l'ordre de chargement : l'ordre de chargement est une impulsion (de largeur égale au temps de stabilisation des adresses) déclenchée par le basculement de 1 à 0 du signal fin de comptage q du circuit de chargement du nouveau point. C'est donc la sortie d'un monostable dont l'entrée est q (monostable à déclenchement sur front descendant).

Circuit de lecture-écriture :

A la réception de l'ordre de chargement, il fournit le signal de chargement. Si on a l'autorisation de lecture-écriture AEL = 0, le signal RAS est activé (RAS = 0). Sinon on attend que AEL soit valide pour activer RAS.
Si pendant un cycle de lecture ou d'écriture il se trouve que la mémoire n'est pas disponible (c-à-d AEL = 1), on arrête le cycle en cours. Dès que AEL passe à 0, on recommence le cycle à l'endroit où il a été interrompu. Pour tout cela, l'ordre de chargement est mémorisé pour former le signal de validation V.
Ce dernier, quand il est actif, permet de relancer le cycle de lecture-écriture, ou de lecture ou d'écriture seulement, quand une interdiction de lecture-écriture se produit pendant un cycle de chargement.
Le signal validation V n'est désactivé que lorsque un cycle est terminé.
Dès réception de l'ordre de chargement, une impulsion S de démarrage de cycle se produit si on a l'autorisation de lecture-écriture ; sinon S l'attend pour se produire. Si au cours d'un cycle de chargement, une interdiction de lecture-

écriture se produit, alors, à la fin de l'interdiction, l'impulsion de démarrage de cycle est générée à nouveau.

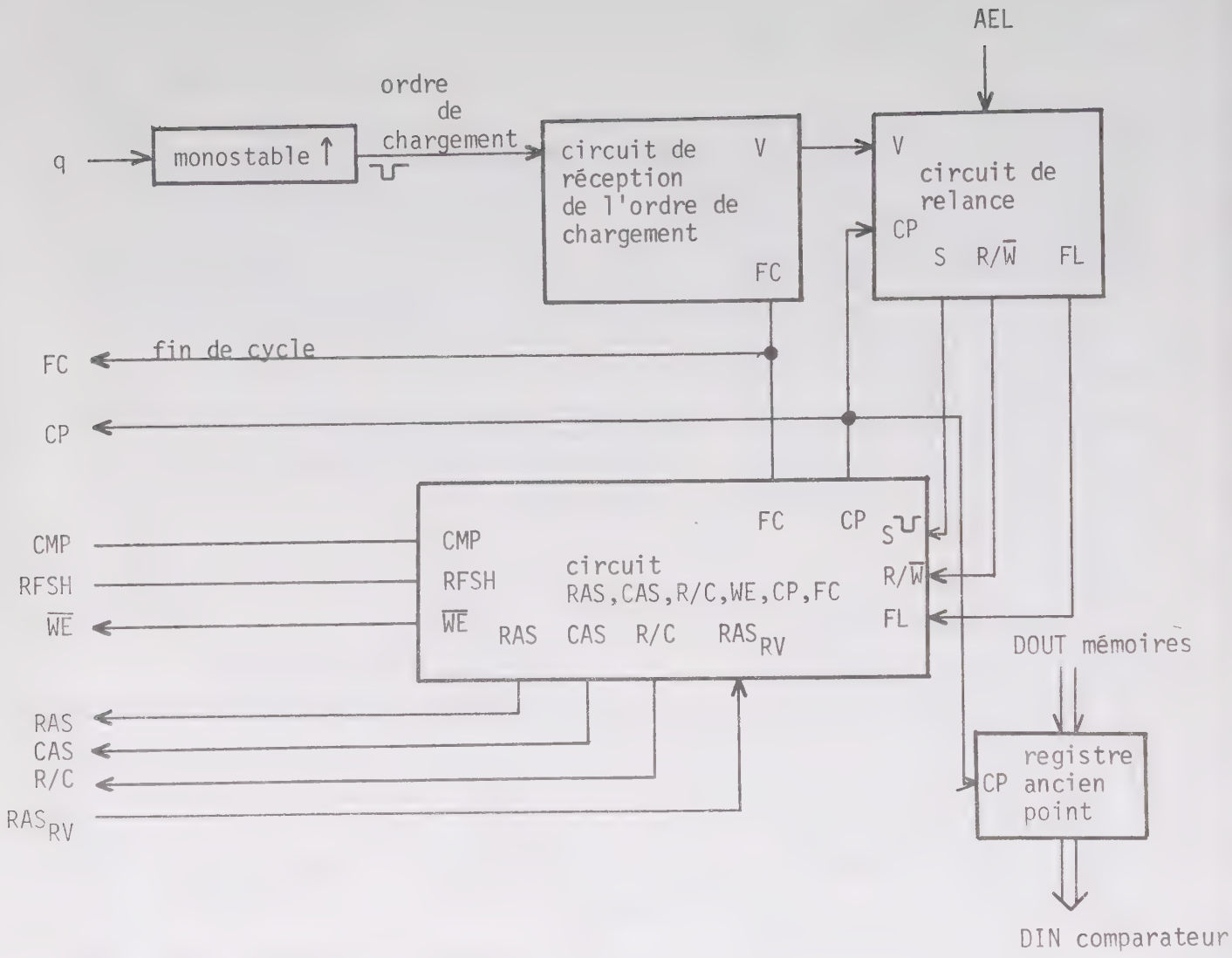
Sur le front montant de l'impulsion de démarrage S : on envoie au circuit d'adressage le signal de multiplexage d'adresse rangée-colonne R/C.
R/C doit rester à 1 (c-à-d que sur le bus d'adresse, on a l'adresse rangée R) pendant un temps de maintien de l'adresse rangée t_{RAH} imposé par la mémoire. Puis il passe à 0 pour qu'on ait sur le bus d'adresse l'adresse colonne. Il repasse à 1 à la fin du cycle ;

: RAS est activé. CAS est activée au bout d'un temps t_{RCD} après l'activation de RAS. $\overline{WE}$ est inactif. Au bout d'un temps t_{CAC} après l'activation de CAS, les données sur la mémoire sont valides. C'est à ce moment que l'on active le signal d'écriture CP du registre ancien point. Alors, on attend le résultat de comparaison. Si le résultat de la comparaison est telle qu'on écrit le nouveau point dans la mémoire d'image, $\overline{WE}$ est activé pendant un temps t_{wp} ($\overline{WE}$ actif à l'état bas). Sur le front montant de $\overline{WE}$, RAS et CAS sont désactivés, R/C retourne à 1 pour qu'au prochain cycle l'adresse rangée soit prête et une impulsion fin de cycle FC est déclenchée, dont le complément logique est l'impulsion de chargement du nouveau point du circuit de chargement du nouveau point.

Si le résultat de la comparaison est telle qu'on écrit l'ancien point dans la mémoire d'image, alors il n'est pas nécessaire de faire un cycle d'écriture, car le point y est déjà. On désactive RAS et CAS, R/C passe à 1 et l'impulsion fin de cycle FC est déclenchée.

Le processus énoncé ci-dessus est le mode normal de fonctionnement à chaque reconnaissance de l'impulsion de démarrage S. Mais il peut arriver, pendant le mode normal de fonctionnement, qu'on n'ait plus l'autorisation de lecture-écriture. Deux cas peuvent se poser : ou le chargement du registre ancien point n'a pas été fait. Alors, dans ce cas, dès la prochaine autorisation de lecture-écriture, le cycle normal est recommencé ;

: ou le chargement du registre ancien point a été fait, mais on n'a pas eu le temps de faire l'écriture. Dans ce cas, un signal $R/\overline{W}$ est activé ($R/\overline{W} = 0$) pour indiquer que le prochain cycle est un cycle d'écriture seulement et $R/\overline{W}$ retourne à 1 à la fin du cycle (seconde partie du cycle normal de fonctionnement).


AEL
ordre de chargement
q
monostable ↑
circuit de réception de l'ordre de chargement
V
FC
V
circuit de relance
CP
S
R/W
FL
FC
fin de cycle
CP
CMP
RFSH
WE
CMP
RFSH
WE
circuit RAS,CAS,R/C,WE,CP,FC
FC
CP
S
R/W
FL
RAS
CAS
R/C
RAS RV
DOUT mémoires
registre ancien point
CP
DIN comparateur
RAS
CAS
R/C
RAS RV

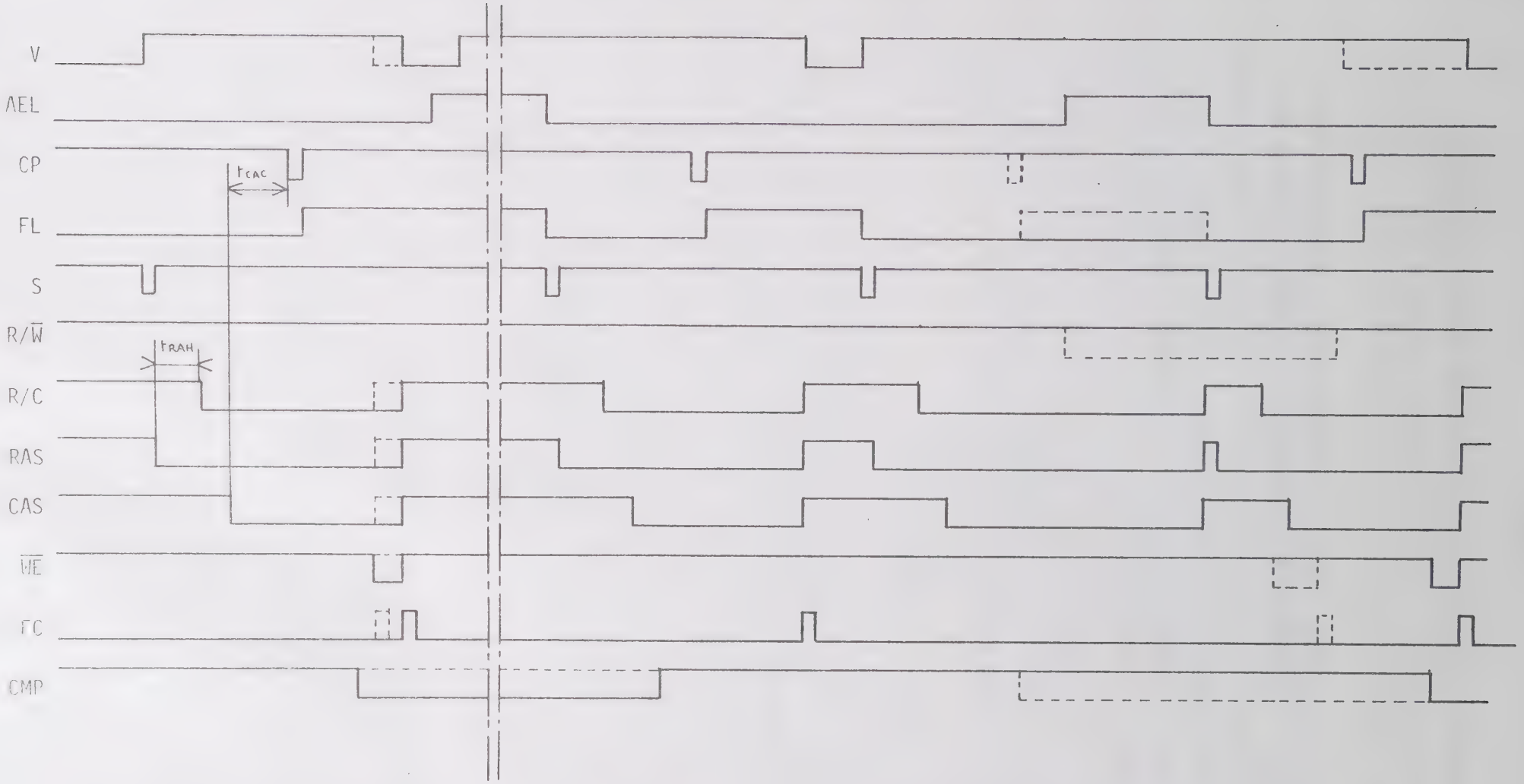


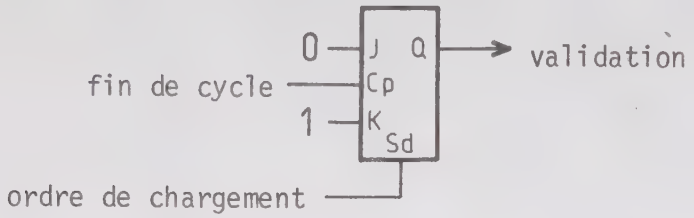
Diagramme des temps du circuit de lecture-écriture

Réalisation du circuit de réception de l'ordre de chargement :

A la réception de l'ordre de chargement, une sortie validation est activée (1). Elle est désactivée (remise à 0) à la reconnaissance d'une impulsion fin de cycle FC.

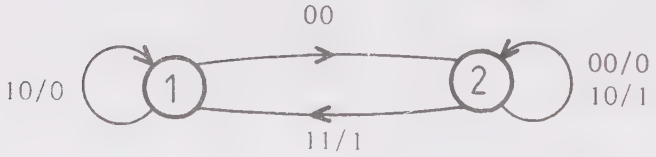
1ère solution : le circuit est réalisé avec une bascule JK dont les entrées J et K sont à 0 et 1. L'ordre de chargement est relié à la borne S de forçage à 1 de la bascule. L'horloge de la bascule est reliée au signal fin de cycle FC.

En effet, dès la réception de l'ordre de chargement, la sortie Q de la bascule JK est mise à 1. Comme JK = 01, alors sur le top d'horloge FC, la sortie est mise à 0.



2ème solution : la seconde solution est réalisée par synthèse de circuit logique séquentiel asynchrone dont le diagramme de fluence est le suivant :

OC FC / V



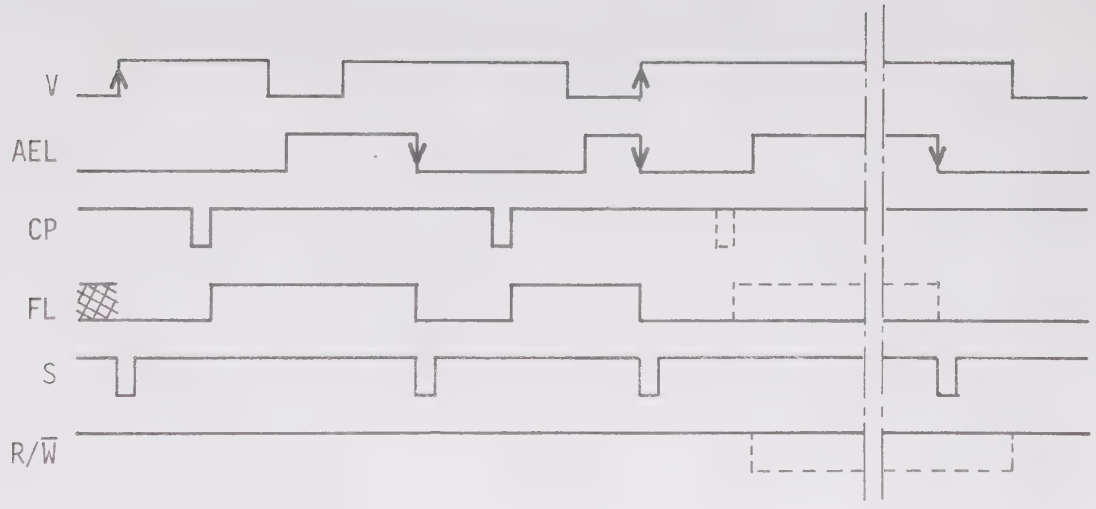
Les équations logiques correspondantes sont :

$$y = \overline{OC} + Y\overline{FC}$$
$$V = OC.Y$$

Réalisation du circuit de relance : il sert à envoyer au circuit d'élaboration des signaux RAS, CAS, R/C, $\overline{WE}$ et CP le top de démarrage S du cycle de lecture-écriture ou de lecture seulement en fonction des signaux V, AEL et CP.

L'impulsion de démarrage S est envoyée à chaque fois que : quand AEL = 0, V passe de 0 à 1 ou quand V = 1, AEL passe de 1 à 0, ou V passe de 0 à 1 et AEL passe de 1 à 0.

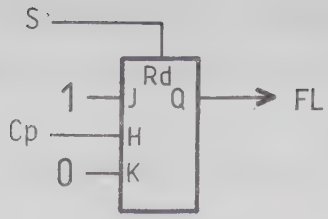
R/$\overline{W}$ passe à 0 quand, pendant un cycle de lecture-écriture (V =1), AEL passe de 0 à 1 alors qu'un cycle de lecture a été fait. De plus, R/$\overline{W}$ passe à 0 pour indiquer au circuit d'élaboration des signaux RAS, CAS, que le prochain cycle est un cycle d'écriture seulement, V restant toujours à 1 (le cycle n'est pas encore terminé). R/$\overline{W}$ retourne à 1 dès que V passe à 0, c'est-à-dire que le cycle est terminé. Pour tout cela on a besoin d'un signal intermédiaire FL, fin de lecture, obtenu à partir de Cp et de l'impulsion de démarrage S.



Réalisation du signal fin de lecture FL :

FL passe à 0 à chaque impulsion début de cycle S et passe à 1 sur le front montant de Cp, indiquant que la donnée a été chargée dans le registre ancien point.

Il est réalisé avec une bascule JK à déclenchement sur front montant. Les entrées J et K sont à 1, l'horloge est Cp et la borne de forçage à 0 Rd est reliée à S.

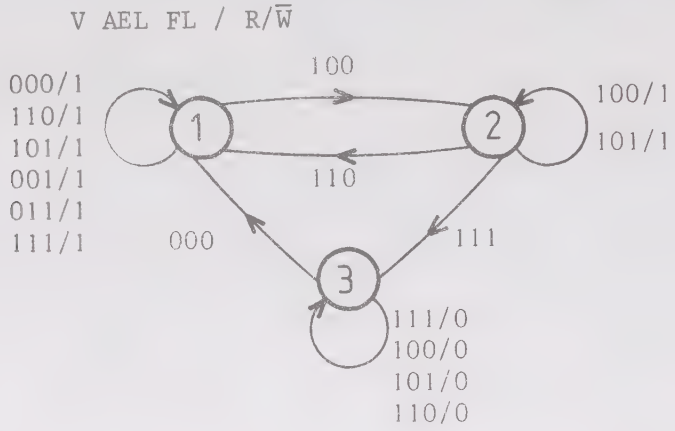


Réalisation du signal $R/\overline{W}$ indiquant la nature du cycle à faire :

Quand $R/\overline{W}$ est à 1, on fait un cycle de lecture-écriture ou de lecture seulement. Quand il est à 0, on ne fait qu'un cycle d'écriture.

$R/\overline{W}$ ne passe à 0 que lorsque FL = 1, V = 1 et que lorsque AEL passe à 1. Il y reste jusqu'au passage de V de 1 à 0.

Le diagramme de fluence correspondant est le suivant :



Les équations logiques sont :

$$y = Y_1 V + Y_2 . AEL . FL$$
$$y = \overline{AEL} . Y + \overline{Y}\ \overline{AEL} . \overline{FL} . V$$
$$R/\overline{W} = \overline{y}_1$$

ce qui donne $R_1 = Y_1 \overline{V}$

avec une $S_1 = Y_2 . FL . AEL$

bascule RS $R_2 = AEL . Y_2$

$S_2 = \overline{Y}_1 . \overline{FL} . \overline{AEL} . V$

Réalisation de l'impulsion de démarrage S :

L'impulsion S se produit au passage de V de 0 à 1 quand AEL = 0, ou sur le passage de AEL de 1 à 0 quand V = 1, ou quand les deux basculements se font en même temps.

1ère solution : elle consiste à fabriquer deux impulsions (l'une sur le front montant de V et l'autre sur le front descendant de AEL) et à les combiner

suivant l'état de V et de AEL. Dans ce cas, on a : S = Sv quand AEL = 0

S = SA quand V = 1

L'expression de S est donc :

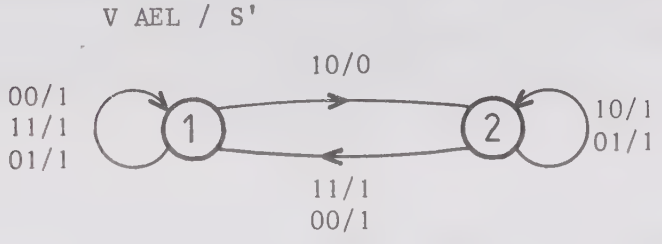
$$S = (Sv + AEL)\ (SA + \overline{V})$$

Dans le cas où AEL et V basculent en même temps, la durée de S est augmentée du temps correspondant au déphasage entre SA et Sv.

Le principe de réalisation de SA et Sv est le même que ceux déjà réalisés.

2ème solution : elle consiste à faire passer S'à 0 pendant les états de transition d'un circuit séquentiel. S est la sortie d'un monostable dont l'entrée est S', car S' risque d'être très courte.

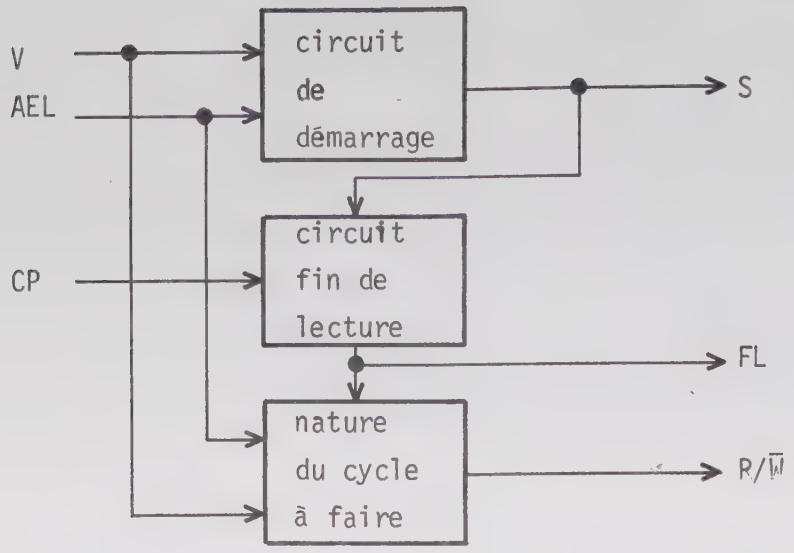
Le diagramme de fluence est le suivant :



Les équations logiques sont :

$y = \overline{AEL}.V + \overline{V}\ AEL.Y$	pour la variable secondaire
$S' = \overline{Y} \uparrow V \uparrow \overline{AEL}$	pour la sortie

Organisation du circuit de relance

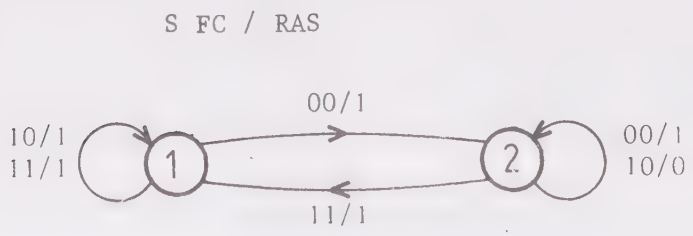


Réalisation des signaux RAS, CAS, R/C, $\overline{WE}$, CP, FC :

Le signal RAS est pris comme signal de référence.

Réalisation de RAS : RAS est à 1 sur le front descendant de S ou sur le front montant du signal fin de cycle FC. Il est activé (RAS = 0) sur le front montant de S.

Le graphe de fluence correspondant est le suivant :



Les équations logiques sont :

$$y = \overline{S}\ \overline{Y} \vee \overline{FC}\ Y \qquad \text{pour la variable secondaire}$$

ce qui donne

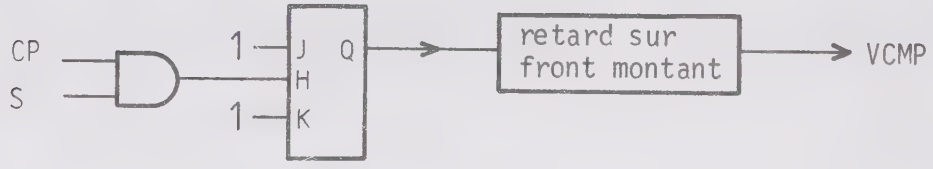
$$J = \overline{S}$$

$$K = FC$$

$$RAS = Y \uparrow S \uparrow \overline{FC} \qquad \text{pour la sortie}$$

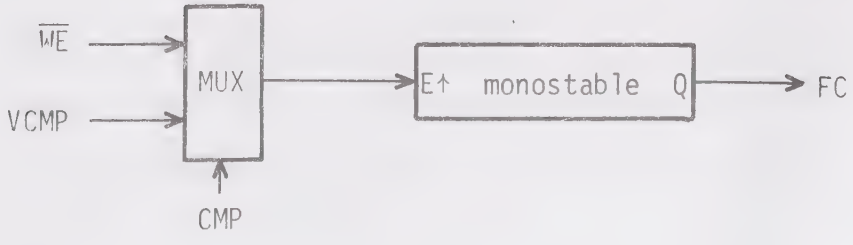
Réalisation du signal fin de cycle FC : l'impulsion FC se produit sur le front montant de $\overline{WE}$ ou sur le front montant d'un signal intermédiaire VCMP (obtenu à partir du front montant de CP quand le résultat de comparaison CMP est au niveau logique 1). [CMP = 1 veut dire qu'on garde dans la mémoire l'ancien point et on n'a plus besoin de faire un cycle d'écriture].

VCMP est au niveau 1 sur le front montant de CP au bout du temps de comparaison. Il est mis au niveau 0 sur le front montant de S. Ceci est réalisé avec une bascule JK dont les entrées J et K sont à 1 et dont l'horloge est S.CP. A la sortie de la bascule, on met un retard égal au temps de comparaison sur le front montant.



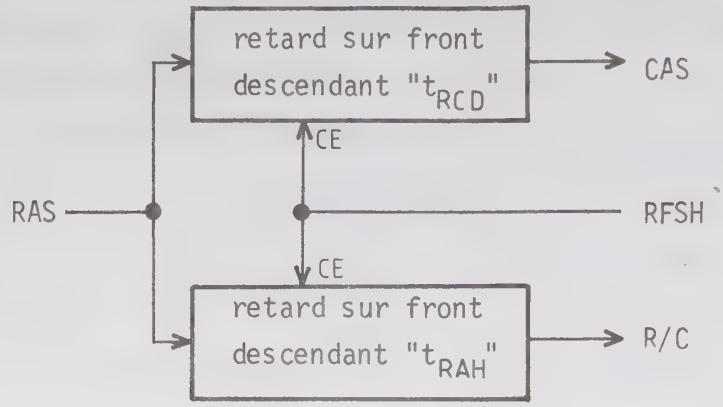
FC est la sortie d'un monostable à déclenchement sur front montant dont l'entrée E est

$$E = \overline{CMP}\ \overline{WE} + CMP.VCMP$$



Réalisation des signaux CAS et R/C : ils sont obtenus à partir de RAS. CAS doit être activé au bout d'un temps t_{RCD} après l'activation de RAS et R/C au bout d'un temps t_{RAH}. Ils sont désactivés (retour au niveau logique 1) en même temps que RAS est désactivé. CAS et R/C doivent être à 1 lors du rafraîchissement des mémoires c-à-d RFSH = 0.

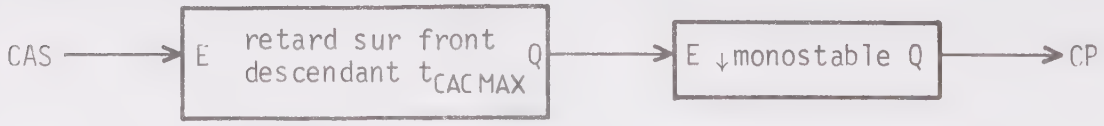
Ils sont donc réalisés par un retard sur le front descendant de RAS. La borne de forçage à 1 du retard est reliée à RFSH.



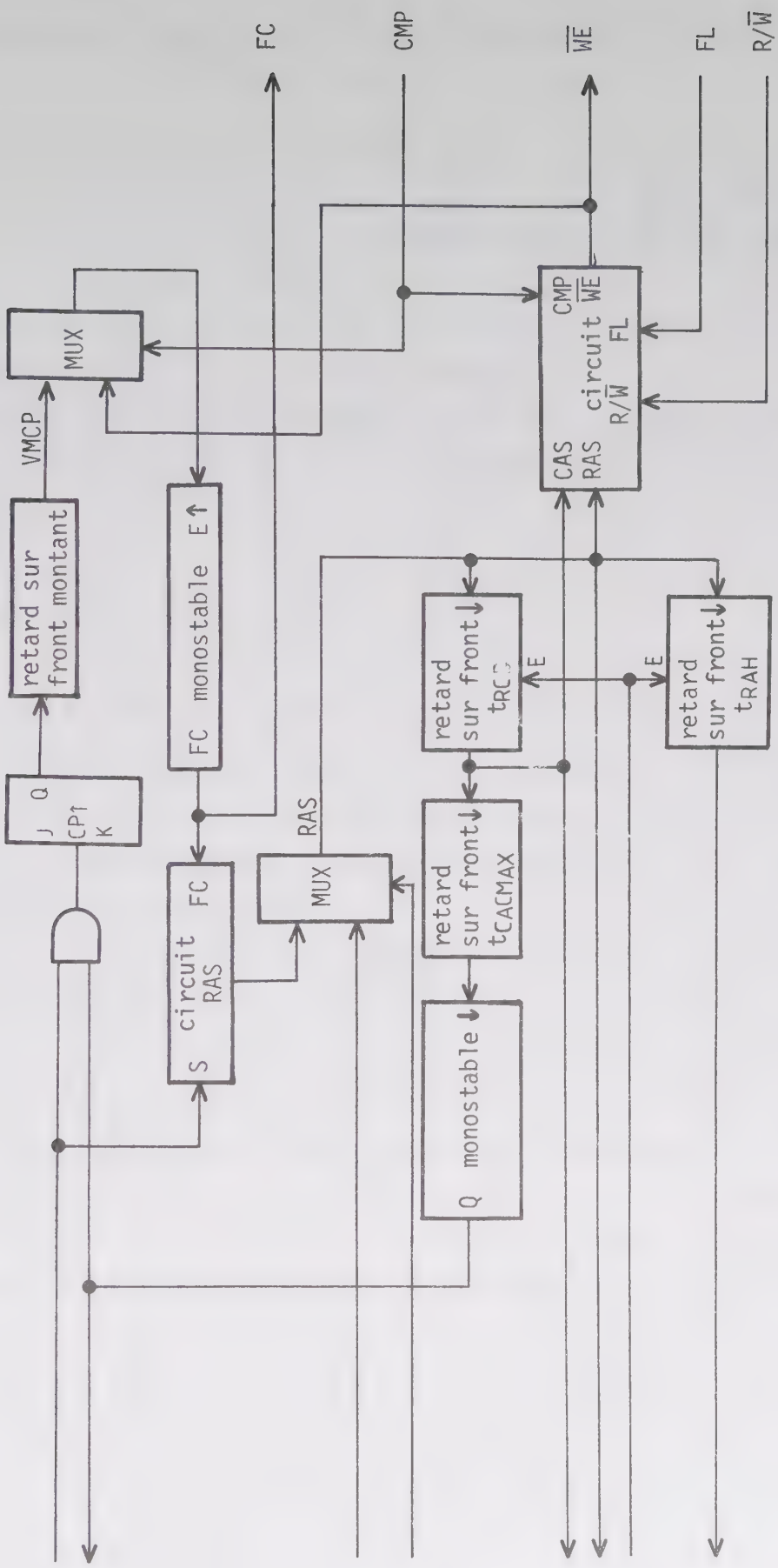
La réalisation du retard sur front descendant est donnée en annexe.

Réalisation du signal d'écriture du registre ancien point CP :

C'est une impulsion qui est déclenchée au bout d'un temps d'accès des mémoires t_{CACmax} après l'activation de CAS. Il est obtenu par un retard de t_{CACmax} sur front descendant de CAS, qui déclenche un monostable à déclenchement sur front descendant.



Réalisation du signal d'écriture mémoire $\overline{WE}$: c'est une impulsion de durée égale au temps d'écriture des mémoires. Elle est déclenchée à la fin de comparaison (c-à-d au bout d'un temps de comparaison t_{CMP} après l'impulsion fin de lecture). Pour cela le signal fin de lecture FL est retardé d'un temps égal à t_{CMP} et sur son front montant est déclenché le signal $\overline{WE}$ quand $R/\overline{W} = 1$ ou sur



ORGANISATION DU CIRCUIT DE REALISATION DES SIGNAUX : RAS,CAS,R/C,$\overline{WE}$,CP,FC

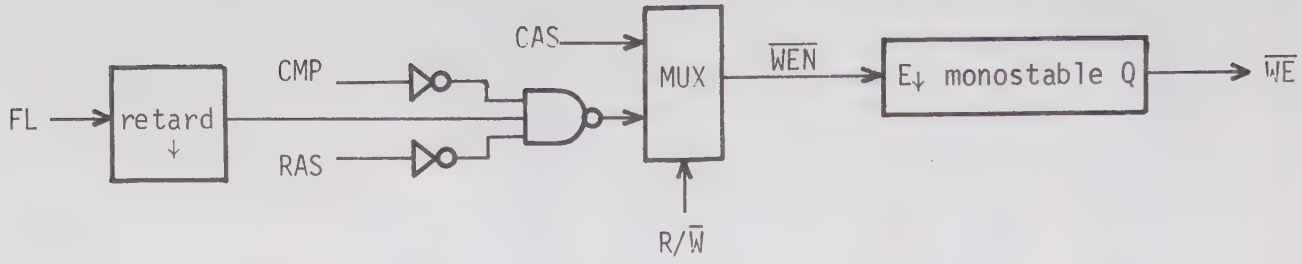
le front descendant de CAS quand $R/\overline{W} = 0$.

On réalise d'abord un signal $\overline{WEN}$ qui déclenche sur son front descendant un monostable de durée twp pour former le signal $\overline{WE}$.

$\overline{WEN}$ = FLr quand RAS = 0 et CMP = 0 et $R/\overline{W} = 1$

$\overline{WEN}$ = CAS quand $R/\overline{W} = 0$

On a donc : $\overline{WEN} = \overline{R/\overline{W}}\ CAS + \overline{\overline{CMP}.FL.\overline{RAS}.R/\overline{W}}$

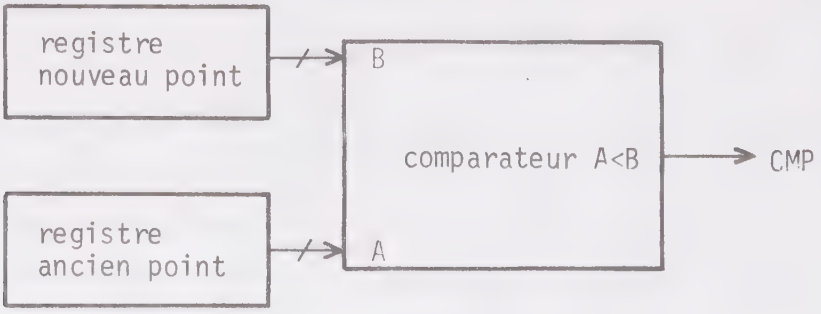


2.5.-Circuit de comparaison :

Il compare les bits de profondeur P_a et P_N contenus respectivement dans le registre ancien point et le registre nouveau point. Si le contenu du registre ancien point est supérieur ou égal au contenu du registre nouveau point, le signal de comparaison CMP est à 0. Il est à 1 dans le cas d'infériorité.

Le comparateur utilisé est un circuit combinatoire, donc asynchrone, pour pouvoir augmenter la vitesse de comparaison, donc augmenter la vitesse de chargement de la mémoire d'image. Il est soit câblé, soit du type 74S85.

En utilisant un comparateur à base de 74S85, la sortie A<B est le signal CMP. Elle est codée de façon à être au niveau 0 en cas d'égalité. L'entrée A du comparateur est le contenu du registre ancien point, et l'entrée B le contenu du registre nouveau point.



2.6.- Circuit de visualisation de l'image et de rafraîchissement des mémoires

Rafraîchissement des mémoires : toutes les n rangées (ici 256) de toutes les mémoires formant chaque zone doivent être rafraîchies à chaque période de rafraîchissement tc qui varie suivant le type de mémoires utilisées. Pour le type de mémoire étudiée, tc est de 4 ms. Il est à noter que, chaque fois que l'on fait un cycle de lecture ou d'écriture ou de lecture-écriture mémoire, toute la rangée de n colonnes est rafraîchie automatiquement. Donc pour une zone, un cycle de visualisation constitue un cycle de rafraîchissement pour les mémoires. Ceci est vrai dans le cas du mode non entrelacé. Pour le mode entrelacé, si la période de visualisation de la zone est supérieure à la période de rafraîchissement mémoire, alors il est nécessaire de rafraîchir les rangées non utilisées (c-à-d celles contenant les lignes paires pendant la trame impaire ou l'inverse), pendant les périodes de suppression lignes. Quelques périodes de suppression lignes suffisent.

L'organisation de la mémoire d'image est conçue pour pouvoir faire une lecture séquentielle et un rafraîchissement automatique des mémoires pendant la période de visualisation en mode non entrelacé. Il n'est donc pas nécessaire de rafraîchir les mémoires pendant cette période. Par contre, en mode entrelacé, il faut un cycle de rafraîchissement particulier pendant cette période.

La période et le mode de rafraîchissement sont les mêmes pour toutes les mémoires formant la zone.

Cas du mode non-entrelacé : le cycle de visualisation est aussi considéré comme un cycle de rafraîchissement ; au bout d'un temps égal à la période de rafraîchissement après la période de visualisation, commence le premier cycle de rafraîchissement des mémoires formant la zone. Ce cycle se répète périodiquement. Il est matérialisé par le signal RFSHi.
Le cycle de rafraîchissement des mémoires de la zone est une lecture de tous les n rangées où CAS = 1, donc RAS est un signal périodique de période égale à la période de lecture mémoire. Pendant le cycle de rafraîchissement, les adresses mémoires sont incrémentées sur le front montant de RAS. Au bout de la dernière rangée, le cycle est arrêté.

Dans le cas où un cycle de visualisation se produit pendant un cycle de rafraîchissement, celui-ci est arrêté et redémarre au bout d'un temps t_R après le cycle de visualisation.

Cas du mode entrelacé : à chaque trame, une partie des rangées n'est pas visualisée. Donc, il est nécessaire de rafraîchir cette partie pendant le

déroulement de la trame.

Suivant la parité de la trame visualisée (supposons que c'est la trame paire), les rangées automatiquement rafraîchies sont donc les rangées contenant les lignes paires (c-à-d les rangées de 0 à 63 et de 128 à 191).

Dès la fin de la visualisation de la première ligne, pendant la période de suppression ligne, on fait un rafraîchissement des rangées non rafraîchies à partir de la rangée 64. Dès le début de la seconde ligne, ce rafraîchissement est arrêté et l'adresse de la prochaine rangée à rafraîchir est mémorisée.

Dès la fin de la seconde ligne, pendant la période de suppression ligne, ce rafraîchissement continue à partir de l'adresse mémorisée et ainsi de suite jusqu'à la fin du rafraîchissement de toutes les rangées non rafraîchies pendant la visualisation de la trame. Ce cycle que l'on vient de décrire recommence au bout de la période de rafraîchissement, si la visualisation n'est pas encore terminée. Pendant le temps où le rafraîchissement n'existe pas, on a une mise à jour de l'image, lors des suppressions lignes. Si la période de visualisation est terminée avant le prochain cycle de rafraîchissement, il est toujours nécessaire de rafraîchir les rangées non rafraîchies.

Pour réaliser ceci, on a donc besoin d'une horloge de rafraîchissement HRFSHi, indépendante pour chaque zone, d'un signal de rafraîchissement RFSHi, du signal RAS et des adresses des rangées à rafraîchir.

Réalisation de l'horloge de rafraîchissement HRFSHi : elle est obtenue par division de la fréquence de lecture f_{RAS}. La fréquence de rafraîchissement est de 500 Hz ou 250 Hz suivant le type de mémoires utilisé.

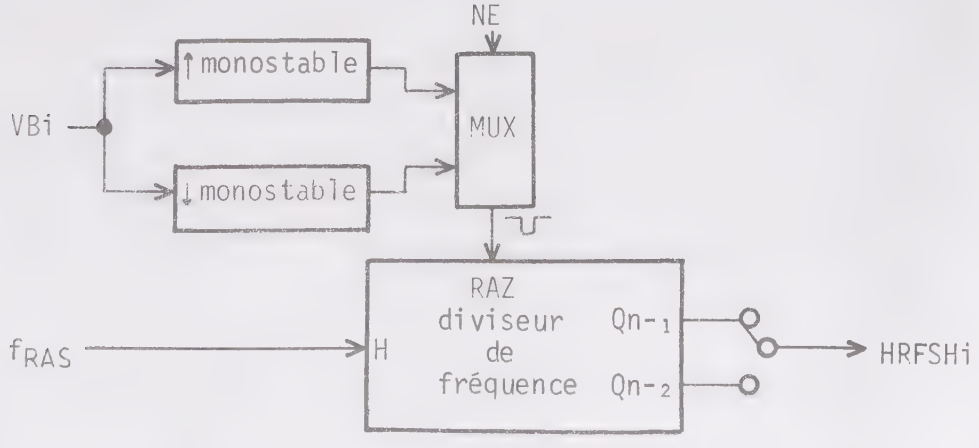
La fréquence de lecture varie aussi suivant le type de moniteur utilisé. Le diviseur de fréquence est remis à zéro à chaque front montant de VBi, en mode entrelacé.

Le tableau ci-après donne le diviseur suivant le type de moniteur utilisé.

Type de moniteur	f_{RAS}	Diviseur 500 Hz	Diviseur 250 Hz
1024 x 1024 non-entrelacé 50 Hz	5,50056 MHz	9061	2
1024 x 1024 non-entrelacé 60 Hz			2
1024 x 1024 entrelacé 50 Hz 60 Hz	2,8125 MHz 3,375 MHz	5620 6744	2
512 x 512 non-entrelacé 50 Hz 60 Hz	2,81 MHz 3,372 Mhz	5620 6744	2
512 x 512 entrelacé 50 Hz 60 Hz	2,5 MHz 2,76 MHz	5000 5520	2
640 x 480 entrelacé 60 Hz	2,76 MHz		2

En mode non-entrelacé, le diviseur de fréquence est remis à zéro sur le front descendant de VBi.

VBi attaque deux monostables qui délivrent chacun une impulsion, l'une sur le front montant de VBi et l'autre sur le front descendant. Les deux impulsions de remise à zéro sont multiplexées suivant le mode.

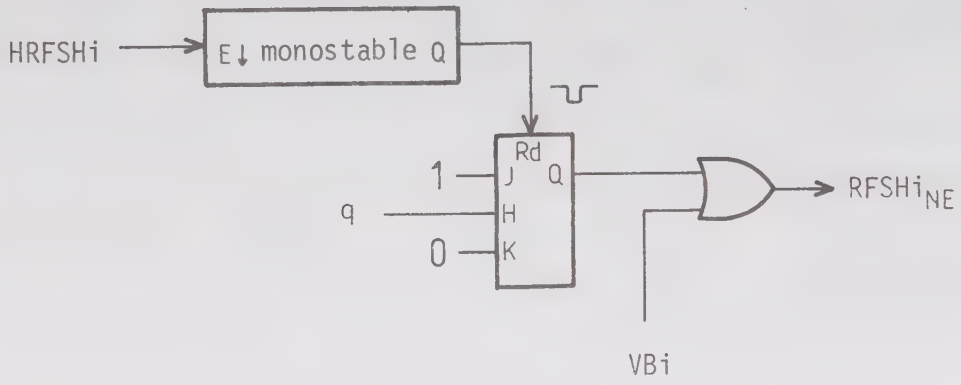


Réalisation du signal de rafraîchissement RFSHi :

Mode non-entrelacé : . en dehors de la période de visualisation (VBi = 0) sur le front descendant de HRFSHi, RFSHi est mis à zéro et retourne à 1 sur le front descendant du signal fin de comptage q de rafraîchissement ;

. pendant la période de visualisation VBi = 1 et RFSHi = 1.

$RFSHi_{NE}$ est donc le OU logique de VBi et de la sortie d'une bascule JK, qui est mise à zéro sur le front descendant de HRFSHi au moyen de la borne de forçage à zéro. Le basculement de q met à 1 cette sortie (front descendant de q).

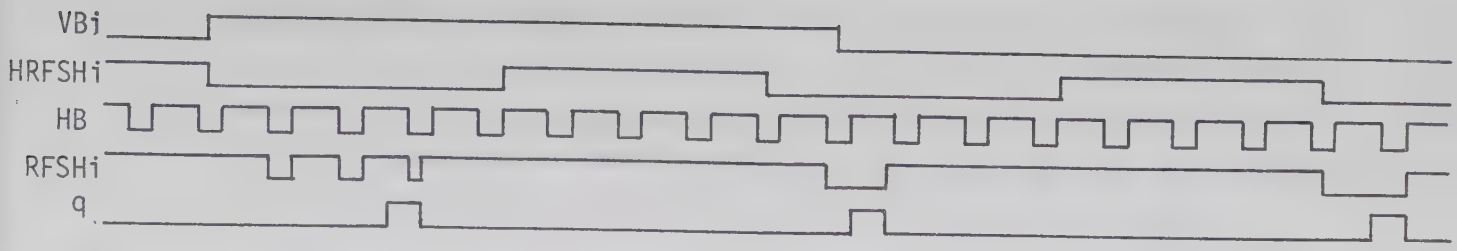


Mode entrelacé ; RFSHi est mis à zéro sur le front descendant de HRFSHi et remis à 1 sur le front descendant du signal fin de comptage du compteur d'adresses, en dehors de la période de visualisation (c-à-d VBi = 0).

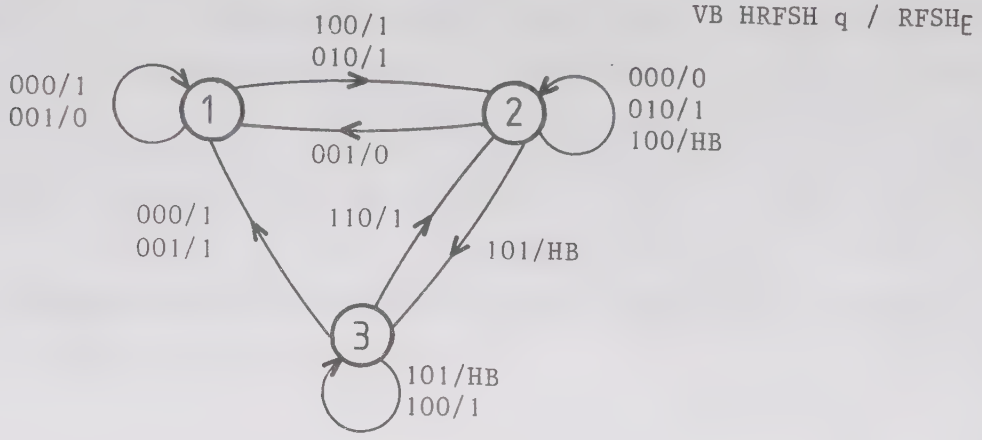
Pendant la période de visualisation (VBi = 1), à chaque front descendant de HRFSHi, RFSHi est égal au signal de suppression ligne HB, jusqu'au front descendant du signal fin de comptage q du compteur d'adresses. Si, au moment où un rafraîchissement est en train de se faire, VBi passe à 1, RFSHi est aussi égal au signal de suppression ligne jusqu'au front descendant de q.

Si, pendant la période de visualisation VBi = 1, il se produit un cycle de rafraîchissement et que la période de visualisation est terminée avant que le rafraîchissement ne soit fini, alors RFSHi continue comme dans le cas où VBi = 0 jusqu'au front descendant du signal q.

Ceci est résumé par le diagramme des temps suivant :



Le diagramme de fluence correspondant est le suivant :



On prend deux sorties Z_1, Z_2 telles qu'elles sont égales à

11 quand RFSH=1

00 " RFSH=0

01 " RFSH=HB

Les équations logiques sont :

pour les variables secondaires

$$y_1 = Y_1\ VB\ \overline{HRFSH} + \overline{Y}_1\ q\ VB$$
$$J_1 = q\ VB$$
$$\overline{K}_1 = VB.\overline{HRFSH}$$
$$y_2 = \overline{Y}_2\ (\overline{Y}_1\ VB + HRFSH) + Y_2\ \overline{q}$$
$$J_2 = \overline{Y}_1\ VB + HRFSH$$
$$\overline{K}_2 = \overline{q}$$

pour les sorties $Z_1 = HRFSH + \overline{Y}_2\ \overline{q}$

$$Z_2 = VB + Z_1$$

et $\boxed{RFSH_E = HB\ Z_2 + Z_1}$ $= H_B V_B + HRFSH + \overline{Y}_2\ \overline{q}$

Réalisation du compteur R : le compteur rangée R est un compteur normal modulo 256 si son entrée MC mode de comptage est à 0. C'est le cas du mode entrelacé en dehors de la période de visualisation ou le cas du mode non-entrelacé. C'est un compteur à déclenchement sur front montant de l'horloge.

Quand l'entrée mode de comptage MC est au niveau 1 (MC = 1), suivant l'entrée type de comptage TC, il compte de 0 à 63 puis de 128 à 191 et revient à zéro quand TC = 0. Quand TC = 1, il compte de 64 à 127 puis de 192 à 255 et revient à 64.

Le problème qui se pose avec ce type de compteur est qu'à chaque fois que TC change de valeur, on saute l'adresse initiale correspondante (0 ou 64). Donc il est nécessaire d'initialiser le compteur à l'adresse initiale correspondante chaque fois que TC change de valeur. Cette initialisation ne peut se faire que si les adresses initiales sont présentes sur le front descendant de l'horloge ou avant.

Les commandes du compteur R sont donc le mode de comptage MC et le type de comptage TC.

MC = 0 mode normal de comptage ∀ TC

MC = 1 mode particulier

Quand MC = 1 et TC = 0, on a un comptage de 0 à 63, puis de 128 à 191 et retour à l'état initial 0.

Quand MC = 1 et TC = 1, on a un comptage de 64 à 127, puis de 192 à 255 et retour à l'état initial 64.

Ce compteur est obtenu par le principe général de synthèse de compteurs à l'aide de bascules. Du fait que le nombre d'étages est faible et que la vitesse du compteur est assez élevée, on choisit un compteur synchrone à report parallèle. La particularité de ce compteur se situe aux deux derniers étages. De l'étage 0 à l'étage 5, on a un compteur synchrone classique, avec

$$J_n = Q_0 Q_1 \ldots Q_{n-1} = K_n$$

Par contre, compte tenu des particularités énoncées ci-dessus, on a :

$$J_6 = Q_0Q_1Q_2Q_3Q_4Q_5\overline{MC} + MC.\overline{TC}$$

$$K_6 = Q_0Q_1Q_2Q_3Q_4Q_5\overline{MC} + MC\ \overline{TC}$$

$$J_7 = K_7 = Q_0Q_1Q_2Q_3Q_4Q_5Q_6\overline{MC} + Q_0Q_1Q_2Q_3Q_4Q_5\overline{Q}_6MC.\overline{TC} + Q_0Q_1Q_2Q_3Q_4Q_5Q_6MC.TC$$

Initialisation du compteur R : le compteur doit être initialisé à chaque mise sous tension ou à chaque changement de type de comptage comme on l'avait énoncé précédemment. L'initialisation est faite de façon asynchrone au moyen des entrées de forçage à 0 ou à 1 des bascules composant le compteur. Il existe deux types d'initialisation : externe ou interne.

Interne : le passage à 1 de TC met Q_6 à 1, $Q_0Q_1...Q_5Q_7$ à 0

" 0 de TC " $Q_6 = 0$, le reste à 0

Externe : quand RESET = 0 ∀ TC MC = 0 $Q_0Q_1...Q_7 = 0$

$Q_0..Q_5Q_7 = 0$

$Q_6 = 1$ si TC = 1 quand MC = 1

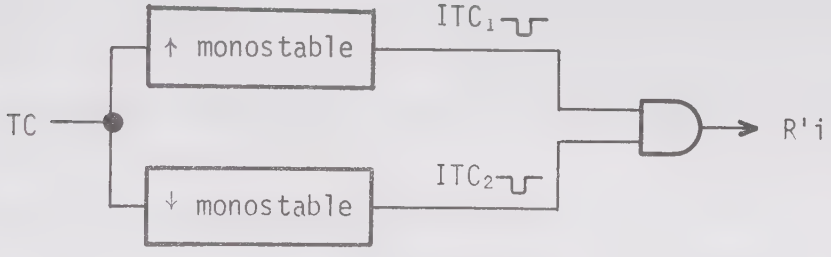
$Q_0..Q_6Q_7 = 0$ si TC = 0 quand MC = 1

on a donc :

$$S_{di} = 1$$
$$R_{di} = \overline{MC}\ RESET + R'i.RESET \text{ pour } i \neq 6$$

où R'i est une impulsion obtenue sur le basculement de TC.

R'i est obtenue de la façon suivante :



pour l'étage 6 on a :

$$Sd_6 = \overline{MC.\overline{RESET}.TC + RESET.\overline{ITC_1}.ITC_2.MC}$$
$$Rd_6 = \overline{\overline{RESET}.\overline{MC} + \overline{RESET}.\overline{TC} + RESET.MC.ITC_1.\overline{ITC_2}}$$

Réalisation des adresses rangées R de rafraîchissement :

Ce sont les sorties $Q_0Q_1..Q_7$ d'un compteur R décrit précédemment.

En mode non-entrelacé, on doit avoir MC = 0 comptage normal, ainsi qu'en mode entrelacé en dehors de la période de visualisation. On a donc :

$$MC = NE.VB_i$$

Quand la trame est paire, on doit rafraîchir les rangées impaires et vice versa. Donc

$$\boxed{TC = \overline{PT}}$$

La validation du comptage est assurée par RFSH.

Quand RFSH = 0, on a H = $\overline{HRAS}$ et quand RFSH = 1, on a H = 1.

On a donc :

$$\boxed{H = RFSH + \overline{HRAS}}$$

Visualisation : suivant l'espace ou l'image choisi, on doit procéder à une lecture générale d'un bloc de 16 mémoires contenant l'image. Dans le type de rangement de mémoires choisi, la lecture des points formant l'image, suivant les deux modes entrelacé ou non-entrelacé, se fait comme suit :

mode non-entrelacé : pendant la période de visualisation, on fait la lecture de toutes les rangées à la cadence de HRAS. A chaque période de visualisation ligne et à chaque fois que la totalité des rangées a été lue, l'adresse colonne est incrémentée et ainsi de suite jusqu'à la $(\frac{256}{z} - 1)^{ième}$ colonne en mode espace, jusqu'à la $(\frac{256}{\text{nb Image}} - 1)^{ième}$ colonne en mode image. Pendant ce temps, les lots non visualisés sont rafraîchis en même temps que la visualisation d'un lot à visualiser se fait.

La période de visualisation est l'ensemble de périodes de visualisation lignes.

En période de visualisation, pendant les suppressions lignes HB = 0, on arrête la lecture et l'adresse de la prochaine lecture est mémorisée. Dès que HB = 1 (période de visualisation ligne). La lecture continue à cette adresse

mode entrelacé : le principe est le même que pour le non-entrelacé, sauf qu'on ne fait que la lecture des rangées contenant les points sur les lignes à visualiser, suivant la parité trame. Pendant la trame paire, on fait la lecture des rangées de 0 à 63 puis 128 à 191. Après 191, l'adresse colonne est incrémentée. Pendant la trame impaire, ce sont les rangées 64 à 127 et 192 à 255.

Dans les deux cas, l'adresse de départ de l'adresse colonne est donnée par les coordonnées du point qui est pris comme origine de l'espace à visualiser. Il doit toujours être un multiple de la définition utilisée.

Réalisation des adresses rangées de visualisation R :

Elles sont obtenues par le même type de compteur R utilisé pour le rafraîchissement. Pour le mode de comptage, on a toujours la même équation :

$$\boxed{MC = NE.VB}$$

Pour le type de comptage, on a l'inverse de celui du compteur R de rafraîchissement, car, quand la trame est paire, on doit rafraîchir les adresses rangées contenant les lignes paires et vice versa. On a donc :

$$\boxed{TC = PT}$$

Le comptage est validé quand HB = 1 et VBi = 1. Dans ce cas, on a $H_v = \overline{HRAS}$. Dans les autres cas, H = 1, ce qui donne :

$$\boxed{H_v = HRAS \uparrow VBi \uparrow HB}$$

Remarque : dans le cas du mode non-entrelacé, on peut utiliser un seul compteur pour la visualisation et le rafraîchissement, car le comptage est normal et les cycles de rafraîchissement et de visualisation sont distincts. Or, dans le cas du mode entrelacé, on ne peut pas avoir un seul compteur. En effet, pendant la période de visualisation, si un cycle de rafraîchissement se produit, les deux cycles sont alternés suivant HB. Quand un cycle est arrêté au profit d'un autre, il faut que le compteur garde sa valeur et puisse partir avec cette valeur dès qu'il aura son tour. Or ceci est impossible avec un seul compteur avec lequel les deux cycles doivent garder leur valeur à leur arrêt, à moins de mémoriser cette valeur par une mémoire auxiliaire et de charger le compteur avant le passage à l'autre cycle.

Réalisation des adresses colonnes de visualisation C_V :

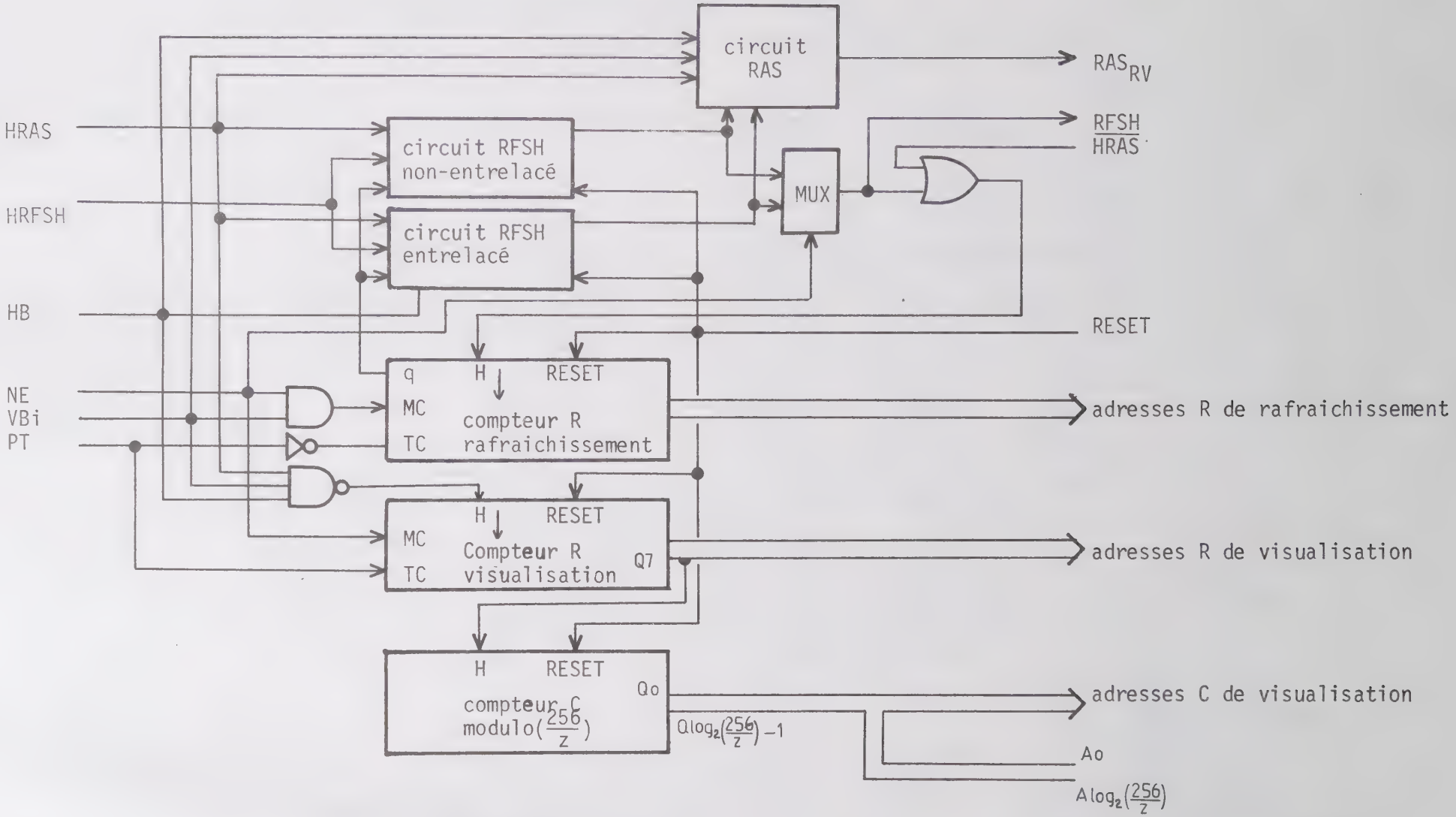
Les $\log_2 (\frac{256}{z})$ premiers bits d'adresse de plus faibles poids sont les sorties d'un compteur d'adresse C modulo $\frac{256}{z}$ et les bits restants sont les bits de la ROMC du circuit d'adressage correspondants. Dans ce cas, les bits reliés à la ROMC ont des valeurs constantes ; initialement on se trouve alors à l'adresse C_o déterminée par ces bits de valeurs constantes. L'adresse C est donc égale à C_o + valeur du compteur C modulo $\frac{256}{z}$. Arrivé à $(C_o + \frac{256}{z} - 1)$, le compteur revient à C_o ; ceci se passe pendant la période de visualisation car en dehors on n'a plus C_o mais C(X,Y) quelconque.

En mode image, pendant la période de visualisation, le procédé reste toujours valable car la ROMC donne toujours l'adresse C_o de l'image à visualiser.

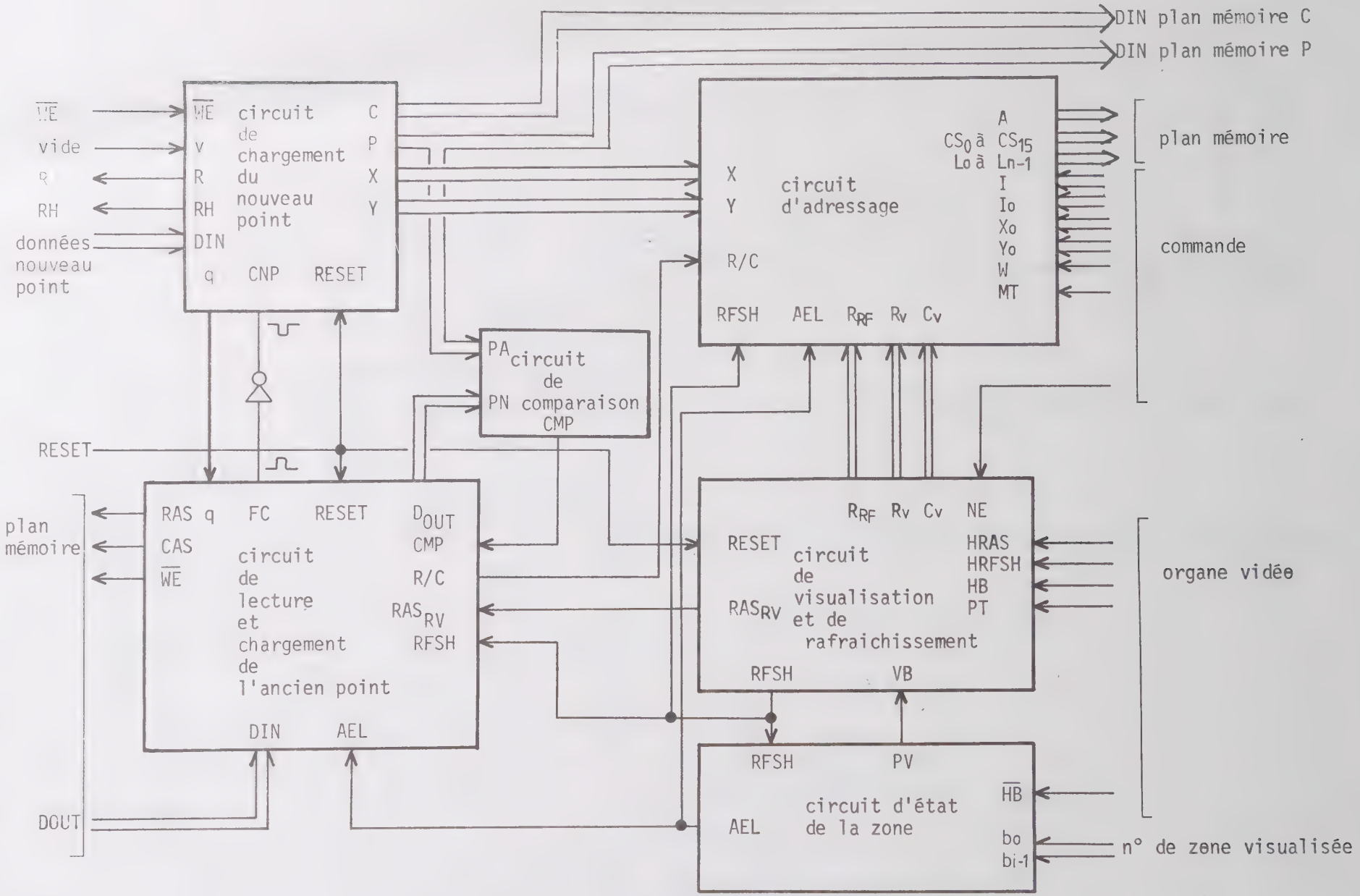
Le compteur C est un compteur normal modulo $\frac{256}{z}$ et le signal d'horloge est la sortie Q_7 du compteur rangée R_V de visualisation.

Réalisation de RAS : le signal RAS est validé, c'est-à-dire égal à HRAS, en période de rafraîchissement (RFSHE = 0 ou $RFSH_{NE}$ = 0) ou en période de visualisation (VB = 1 et HB = 1). Ce qui donne :

$$RAS_{RV} = \overline{VB.HB}.RFSH_{E}.RFSH_{NE} + HRAS$$



ORGANISATION DU CIRCUIT DE VISUALISATION DE L'IMAGE
ET DE RAFRAICHISSEMENT DES MEMOIRES



ORGANISATION DU CIRCUIT DE MISE A JOUR DE L'IMAGE

2.7.- Séquenceur :

Il doit envoyer à toutes les zones le numéro de la zone en train d'être visualisée pendant la période de visualisation générale (VB = 1). Etant donné que la période de visualisation est un multiple de la période ligne, le numéro de zone visualisée peut être obtenu par un compteur modulo le nombre de zones et dont l'horloge est un multiple de la période ligne pour un nombre de zones z donné. Si Dv est la définition verticale, on a $tv(z) = \frac{Dv}{z} H$.

L'horloge du compteur zone est donc la $\log_2(\frac{Dv}{z})^{ième}$ sortie d'un diviseur de la fréquence horizontale par $\frac{Dv}{z}$.

Exemple : soit Dv = 1024 lignes

$z = 4$

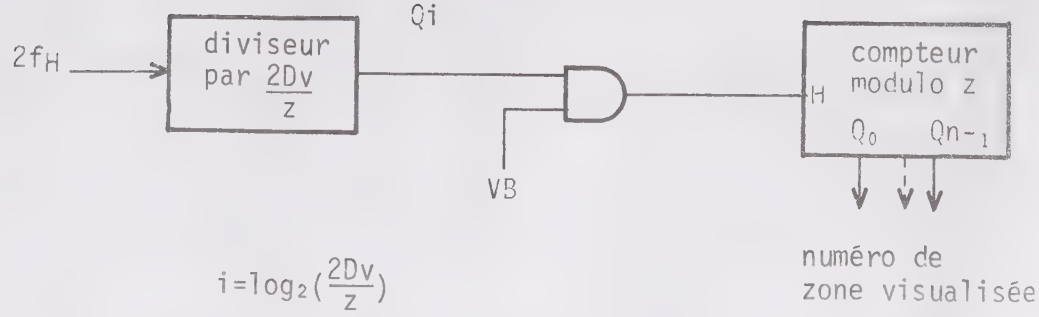
alors $\frac{Dv}{z} = \frac{1024}{4} = 256$ lignes par zone.

La durée d'une zone est donc de 256 H. Donc, toutes les 256 H, le compteur modulo z doit s'incrémenter. L'horloge est la $9^{ième}$ sortie du diviseur de fH par 256.

Quand VB = 1 (période de visualisation), le compteur modulo z fonctionne et quand VB = 0 (période de suppression trame), le compteur modulo z est inhibé.

Soit Q la sortie égale à $\log_2 \frac{Dv}{z}$ on a :

$$H = Q.VB$$



Les étages du diviseur par $\frac{Dv}{z}$ peuvent être ajoutés ou enlevés suivant le nombre de zone désiré.

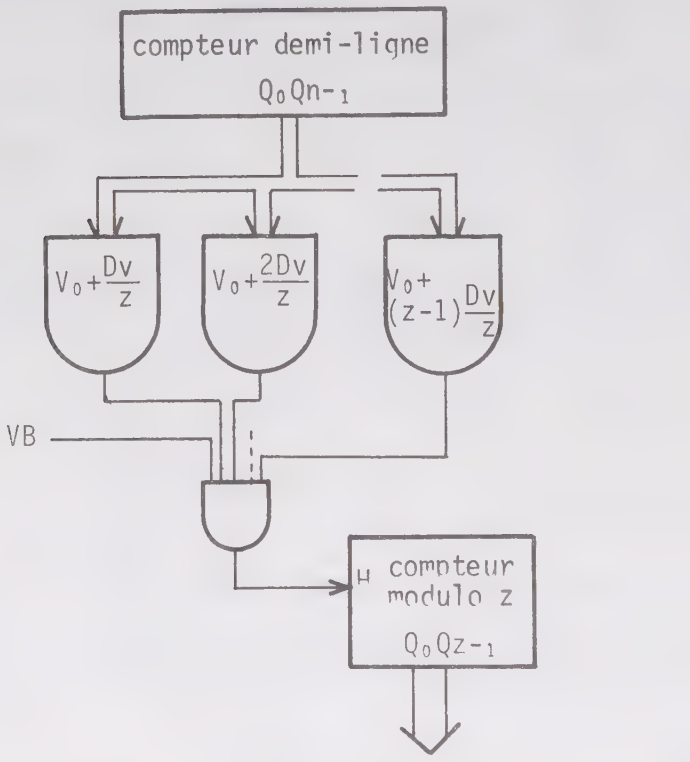
Une autre solution consiste à employer le compteur demi-ligne utilisé dans l'organe vidéo décrit plus loin, ce qui permet de s'affranchir du diviseur par $\frac{Dv}{z}$. Dans ce cas, on utilise des décodeurs à base de porte ET ou NON ET reliées

à la sortie du compteur demi-ligne. Ces décodeurs délivrent une impulsion de largeur égale à la période demi-ligne. A chaque impulsion délivrée par les décodeurs, le compteur modulo z s'incrémente.

Soit : Vo la valeur du compteur demi-ligne à laquelle VB est mis à 1
Dv la définition verticale
z le nombre de zones.

Les décodeurs doivent reconnaître les valeurs $Vo + \frac{Dv}{z}$, $Vo + \frac{2Dv}{z}$ ... $Vo + (z-1)\frac{Dv}{z}$.

Si les impulsions sont négatives alors, pour avoir l'horloge du compteur modulo z, on fait un ET de tout ces décodeurs.



Cette solution a l'avantage de pouvoir être utilisée en mode non-entrelacé ou en mode entrelacé. En effet, en mode entrelacé, une trame sur deux doit commencer sur une demi-ligne. Ceci ne peut être utilisé avec l'autre solution à moins d'utiliser 2 fH comme horloge et d'utiliser un diviseur de $\frac{2Dv}{z}$. Dans ce cas, c'est l'autre qui aura l'avantage car quand z est grand on aura (z-1) décodeurs pour celui-ci et pour l'autre $\log_2 \frac{Dv}{z}$ étages. La solution adoptée est donc la première.

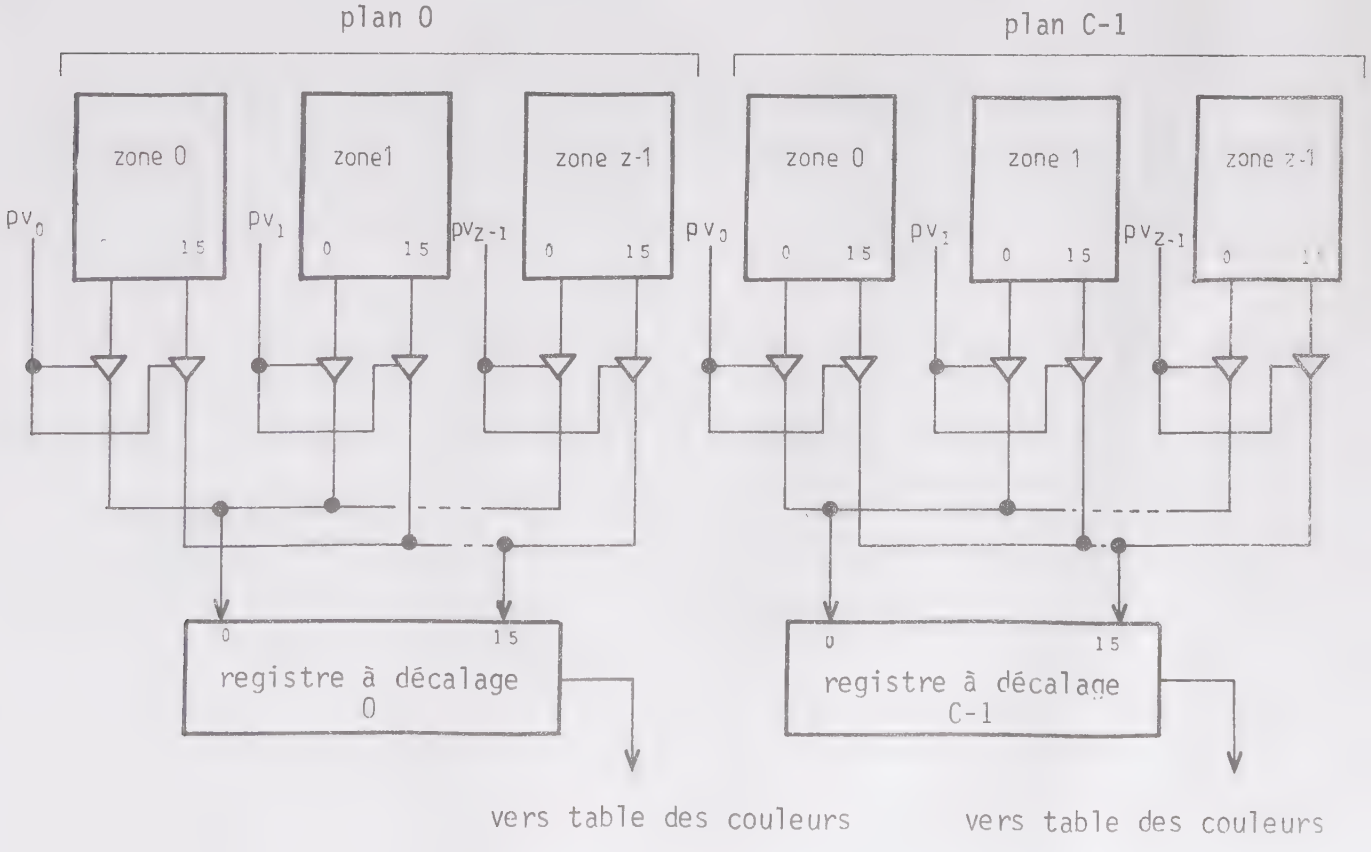
B. ORGANE VIDEO

3.- Conception du circuit de rafraîchissement de l'image :

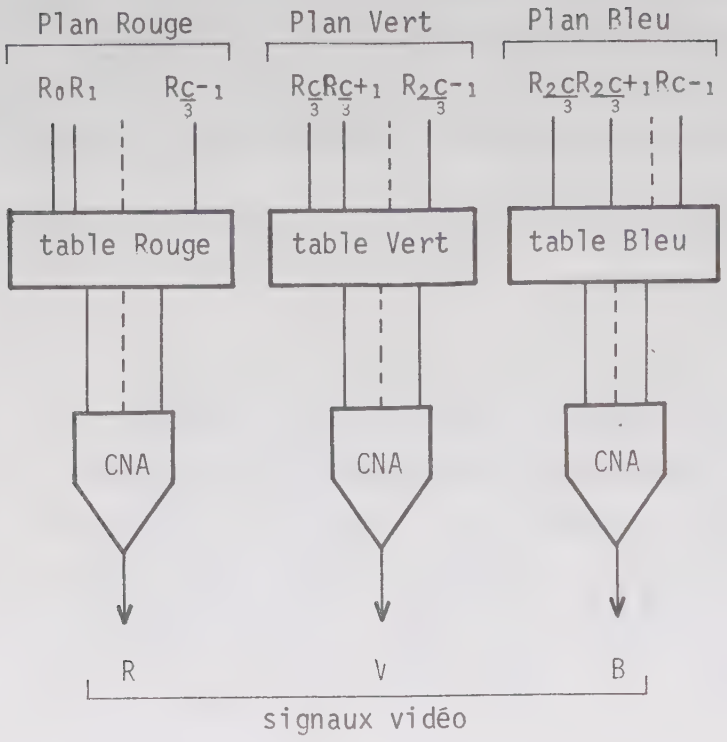
Pour C bits de couleurs, on doit avoir C plans mémoires de zones indépendantes. Le circuit est composé de C registres à décalage correspondant aux C bits de couleurs (trois tables de couleurs et trois convertisseurs numériques analogiques pour chacun des trois couleurs primaires R,V,B). Les C bits de couleurs sont divisés en trois.

Les $\frac{C}{3}$ premiers bits sont pour le Rouge, les $\frac{C}{3}$ suivants pour le Vert et les $\frac{C}{3}$ derniers pour le Bleu.

Supposons que l'on ait 16 bits de couleur. Les 16 bits de chaque zone passent chacun à travers un circuit à trois états avant d'aller au registre à décalage. Les sorties des circuits à trois états, dont les bits correspondants sont reliés ensemble comme indique la figure, sont sélectionnés suivant le signal de visualisation pv_i de chaque zone. Seule la zone en période de visualisation communique donc avec le registre à décalage. Ceci permet de réduire le nombre de registres à décalage, tout en évitant un conflit de bus de données des zones.



Les sorties des registres à décalage vont aux tables de couleurs comme l'indique la figure suivante :



Les tables des couleurs sont des mémoires vives (RAM) rapides dont les adresses sont les sorties des ensembles de registres à décalage correspondant à chaque primaire. On dispose de $2^{\frac{C}{3}}$ niveaux pour chaque primaire, qui sont choisis parmi les 2^n niveaux possibles si n est le nombre de bits de données des tables de couleur.

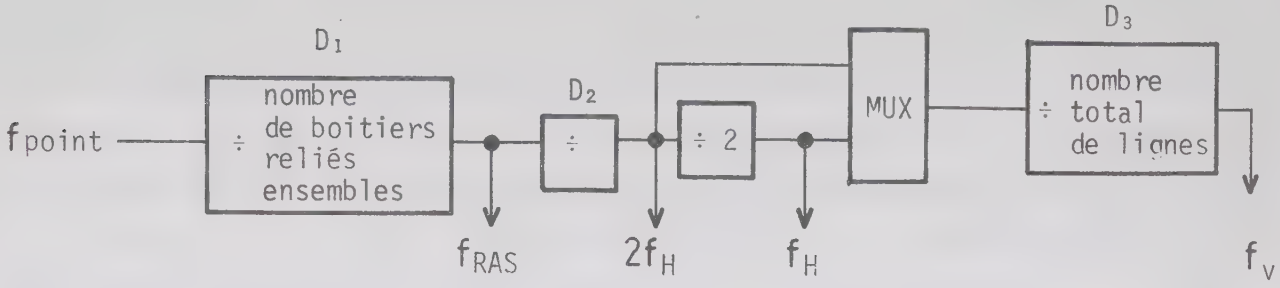
Exemple : supposons qu'on ait 2 bits pour chaque primaire (c-à-d 6 bits de couleurs) et que la table de couleurs ait 3 bits de données.

On peut donc avoir 2^6 couleurs possibles parmi 512. Dans chaque échelle, on a 4 niveaux parmi 8.

Le chargement des tables de couleurs peut être fait dynamiquement en dehors des périodes de visualisation, c'est-à-dire pendant la période de retour trame. Il est fait extérieurement.

4.- Conception du générateur de signaux de synchronisation :

Les horloges de synchronisation, telles que HRAS pour la réalisation de RAS, la fréquence horizontale et la fréquence verticale, sont obtenues par divisions successives de l'horloge de base à la fréquence point.



Choix de la fréquence point : elle est fonction du type de moniteur utilisé, du type de standard utilisé.

Un moniteur télévision est caractérisé par :

- la fréquence verticale fv
- la fréquence horizontale fh
- le taux d'utilisation de la ligne α_L
- le taux d'utilisation de l'image α_i
- le nombre total de lignes L

Soit : H la période ligne $H = \frac{1}{fh}$

V la période image $V = \frac{1}{fv}$

Lu durée de ligne utile $Lu = \alpha_L H$

Vu durée d'image utile $Vu = \alpha_i V$

Dy la définition horizontale, nombre de points visibles par ligne

La fréquence point fp est le rapport de la durée de ligne utile et de la définition horizontale.

Applications :

- Haute résolution 1024x1024 non-entrelacé :

1066 lignes noir et blanc

Lu = 11,9 µs

Vu = 16 ms

fh = 63,96 KHz $\Rightarrow$ H = 15,63 µs

fv = 60 Hz

on a donc $Hp = \frac{11,9\ \mu s}{1024} \Rightarrow fp = \frac{1024}{11,9\ \mu s} = 86,05$ MHz

La fréquence point est choisie autour de la fréquence trouvée ci-dessus, de façon à ce que le second diviseur de fréquence soit un diviseur par un nombre entier.

Une fréquence inférieure risquerait d'entraîner le fait que certains points ne seraient pas visibles. La fréquence supérieure est limitée par la bande passante du moniteur.

Le problème revient donc à trouver fp de façon à ce que D_2 soit un entier. Alors, dans ce cas, fp = 88,00896 MHz et D_2 = 43.

- Haute résolution Noir et Blanc ou couleurs : 1024x1024 entrelacé :

Couleur	Noir et Blanc
40 Hz $\leqslant$ fv $\leqslant$ 70 ou 90 Hz suivant les constructeurs	50 Hz $\leqslant$ fv $\leqslant$ 60 Hz
28 KHz $\leqslant$ fh $\leqslant$ 35 KHz	

Dans cette gamme, le nombre de lignes et la fréquence de balayage ne sont plus fixés par le constructeur, mais au gré de l'utilisateur, tout en tenant compte de la relation entre le nombre total de lignes et les fréquences de balayage.

Il est à noter que, en mode entrelacé, le nombre total de lignes doit être impair du fait du décalage d'une demi-ligne d'une trame par rapport à l'autre.

Pour ces moniteurs, quelque soient les fréquences de balayage utilisées et le nombre de lignes utilisées, le temps de suppression trame minimum VB_{min} est de 27 H. Il faut donc, pour avoir une définition verticale de 1024 lignes, que (27H+512H+0,5)x2 = 1079 lignes minimum.

On a :

fp = 45 MHz et D_2 = 50 pour fv = 50 Hz et nombre de lignes = 1125

fp = 54 MHz et D_2 = 50 pour fv = 60 Hz " " = 1125

On a exactement la même chaîne diviseur pour une fréquence balayage verticale de 60 Hz ou de 50 Hz. Donc pour passer d'une fréquence de balayage à une autre, il suffit de changer la fréquence de l'horloge point.

- Moyenne résolution Noir et Blanc ou couleurs 512x512 non-entrelacé :

On emploie le même type de moniteur utilisé pour le 1024x1024 entrelacé. Dans ce cas, on n'utilise qu'une trame sur deux qui est répétée 50 ou 60 fois par seconde suivant le cas.

fp = 22,48 MHz, D_2 = 50 pour fv = 50 Hz et nombre de lignes = 562
fp = 26,976 MHz, D_2 = 50 " = 60 Hz " " = 562
nb de boîtiers réunis ensemble = 8

- Moyenne résolution Noir et Blanc ou couleurs 512x512 entrelacé :

Dans cette résolution, il existe deux types de moniteur : les moniteurs standard moyenne résolution et les téléviseurs de commerce (sauf pour le standard 60 Hz). Pour cette résolution, on utilise un bloc de 8 mémoires au lieu de 16.

Moniteurs standards : 40 Hz $\leqslant$ fv $\leqslant$ 70 Hz
15 KHz $\leqslant$ fh $\leqslant$ 184 Hz

Avec ces moniteurs, on ne peut pas utiliser la même chaîne division pour les fréquences de balayage vertical de 50 Hz et de 60 Hz car le nombre de lignes minimum pour une fréquence de balayage de 50 Hz est le même que le nombre de lignes maximum pour une fréquence de balayage de 60 Hz
$\frac{18\ KHz}{30} = 600$ et $\frac{15\ KHz}{25} = 600$. Or ce nombre est pair, ce qui ne convient pas pour un balayage entrelacé.

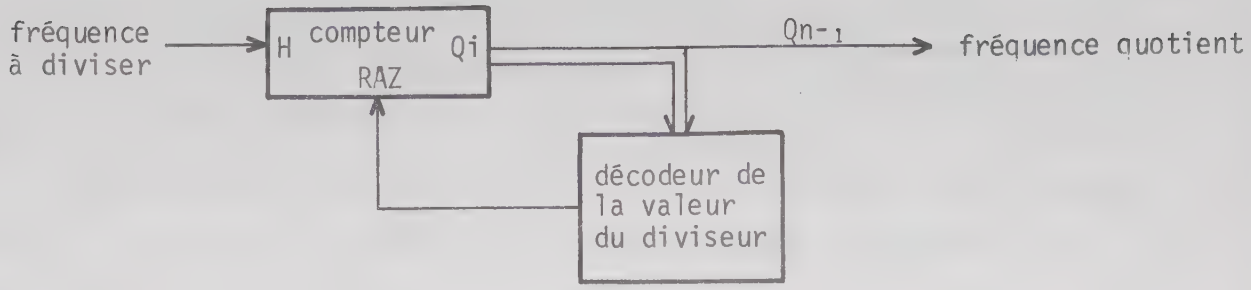
On a donc : fp = 10 MHz D_2 = 80 pour fv = 50 Hz Nbre de lignes = 625
fp = 11,04 MHz D_2 = 80 pour fv = 60 Hz " " = 575
téléviseurs de commerce : on a la même chose pour 50 Hz, mais le standard est de 525 lignes à 60 Hz et la résolution est de 640x480.
On a donc : fp = 12,6 MHz D_2 = 50 pour fv = 60 Hz Nbre de lignes = 525.

Réalisation des diviseurs de fréquence :

Les diviseurs de fréquence sont réalisés à partir de compteur asynchrone qui est remis à zéro à la reconnaissance de la valeur du diviseur. La première division est composée de $\log_2(n)$ étages car le diviseur est une puissance de deux.

Pour les diviseurs qui sont différents d'une puissance de deux, on fait un décodage de la valeur du diviseur. A chaque fois que le contenu du compteur est égal au diviseur, on fait une remise à zéro du compteur. Le temps de remise à zéro du compteur étant égal au temps de propagation du décodeur ajouté au temps de remise à zéro du compteur, il est de l'ordre de 50 ns. Comme la sortie du diviseur est le dernier étage du compteur, pour elle, aux fréquences de travail, le temps de remise à zéro est pratiquement invisible, s'il est très petit par rapport à la période de l'horloge à diviser. Cette condition est vérifiée à partir du second diviseur.

L'organisation du diviseur de fréquence est la suivante :



Principe de l'ensemble mémoire d'image et registre à décalage :

Ce principe conduit à la réalisation de l'horloge HRAS et du signal de chargement des registres à décalage. Il est basé à partir des mémoires RAM 64K dynamiques utilisées pour la haute résolution 1024x1024 non-entrelacé qui nécessite un temps de lecture des mémoires rapide et dont le rapport cyclique du signal RAS est fonction des mémoires utilisées, de l'ordre de 60%. Une fois que la version rapide est réalisée, il va de soi pour la version lente (1024x1024 entrelacé et le reste).

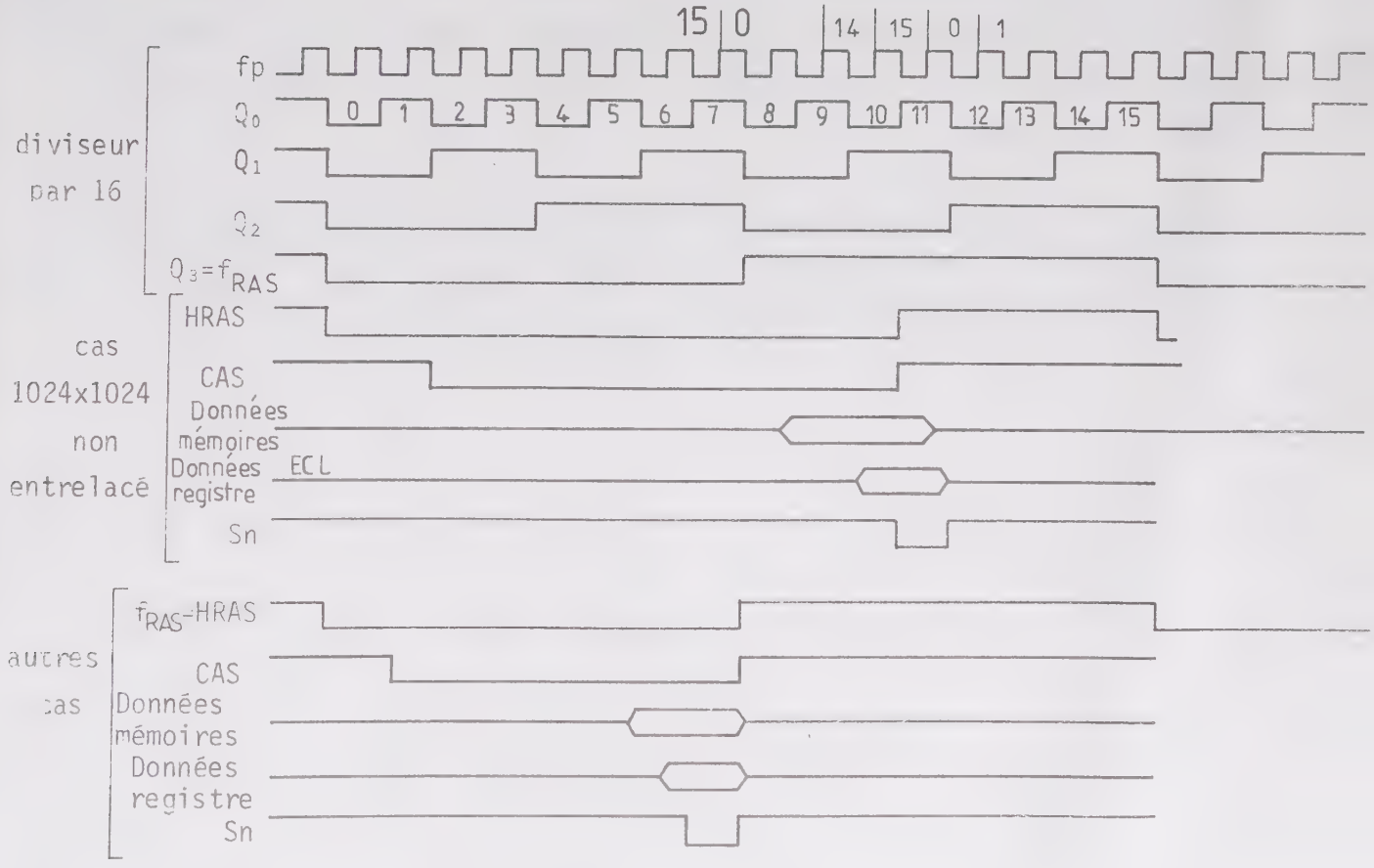
Le cycle de lecture d'un bloc de 16 mémoires débute sur le front descendant de fRAS où HRAS est validé (niveau bas) et y reste pendant 10 périodes d'horloge point puis retourne au niveau haut à la 12ième période d'horloge et y reste pendant 5 périodes d'horloge, ce qui fait un rapport cyclique de $\frac{5}{11}$, car un cycle de lecture mémoire dure 16 périodes d'horloge point.

Quand RAS est validé, les données sont valides au bout d'un temps d'accès de la mémoire qui est variable suivant le type utilisé et qu'on ne peut contrôler. Comme la période d'horloge point est du même ordre de grandeur que le temps de propagation des signaux TTL, il faut tenir compte de ce temps de propagation et aussi du temps de conversion TTL, ECL dans le cas de la haute résolution 1024x1024 non-entrelacé (dont la fréquence point est en dehors de la limite d'utilisation des portes TTL). Dans les autres cas, il ne faut tenir compte que du temps de propagation.

Les registres sont décalés sur le front montant de l'horloge point. Le signal de chargement du registre se fait sur le front montant de HRAS, car compte tenu du temps de propagation des signaux, c'est à ce moment là que les données sont stables au niveau du registre. Il revient au niveau 1 dès le prochain front descendant de l'horloge point, c'est-à-dire pendant le 11ième top d'horloge

point à partir de la validation de RAS. Pendant l'impulsion de chargement sur le front montant de l'horloge point, on a chargement du registre à décalage ce qui assure une continuité du décalage. En effet, supposons que la sortie du registre à décalage soit le point n° 15, il était là sur le front montant de l'horloge point. Sur son front descendant, on a validation du signal de chargement du registre à décalage, mais ce chargement ne se fait que sur le front montant de l'horloge point. Donc, au prochain front montant de l'horloge point, on a chargement du registre à décalage. Ceci est équivalent à un décalage point car avant, on avait le $(n-1)^{ième}$ point sur Qn (Qn est la dernière sortie du registre à décalage). Au prochain front montant, on a à la sortie Qn le $n^{ième}$ point obtenu par chargement et au top suivant (front montant), le $(n+1)^{ième}$ point par décalage.

Ce principe est résumé par le diagramme de temps suivant :



Réalisation du signal de chargement Sn des registres à décalage :

cas du 1024x1024 non-entrelacé : pendant le 11ième top d'horloge, Sn doit être activé (actif à l'état bas). On a donc :

$$\boxed{Sn = \overline{Q}_0 + \overline{Q}_1 + Q_2 + \overline{Q}_3}$$

la logique utilisée est la logique ECL.

autres cas : logique TTL. L'écriture des registres se fait au milieu du cycle RAS, donc à la reconnaissance de la valeur 7. D'où :

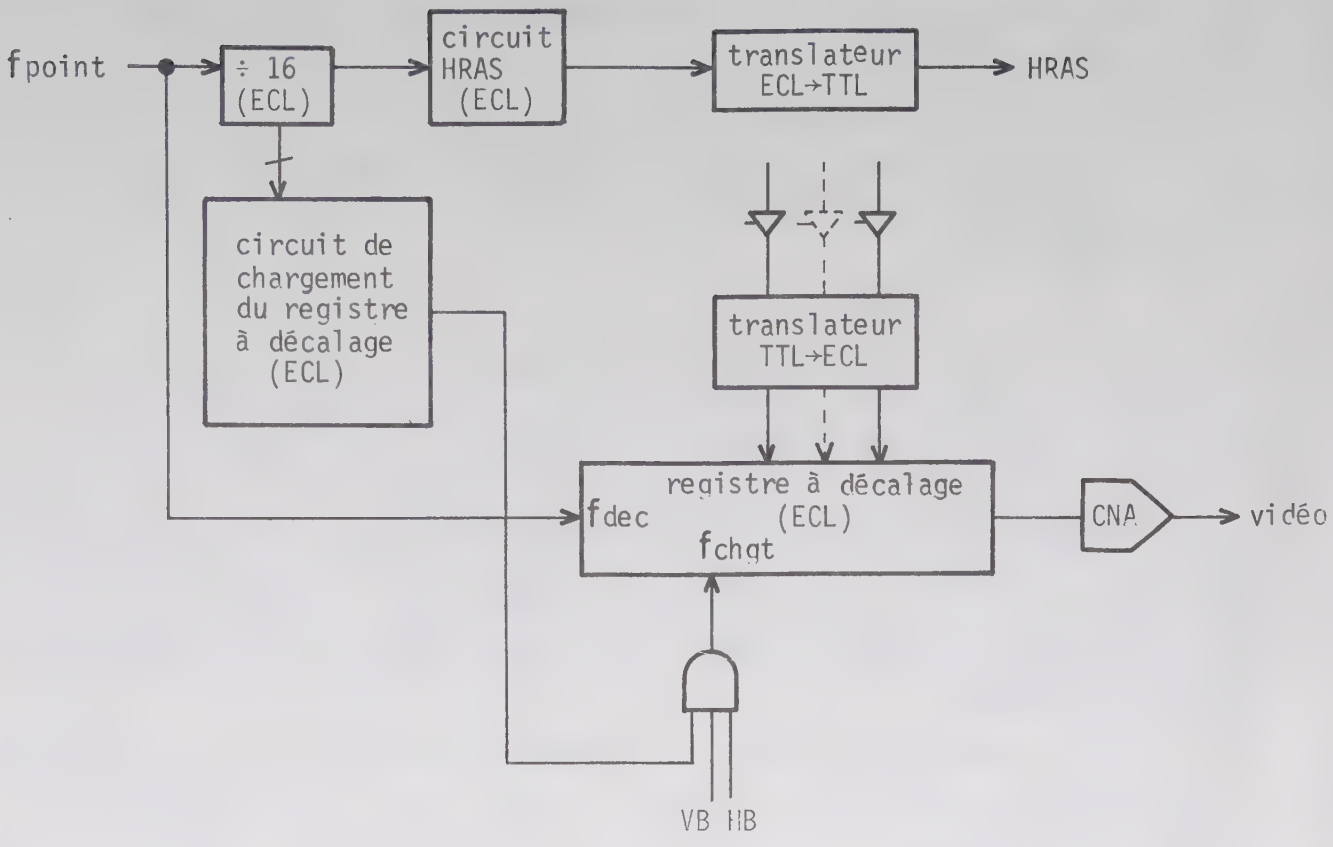
$$\boxed{Sn = Q_0 \uparrow Q_1 \uparrow Q_2 \uparrow \overline{Q}_3}$$ quand VB et HB sont au niveau 1

Réalisation de HRAS :

autres cas : on a HRAS = fRAS

cas du 1024x1024 non-entrelacé : HRAS est activé sur le front descendant de fRAS et désactivé sur le front descendant de Sn. On utilise des circuits ECL pour le diviseur par 16 de la chaîne division, pour le circuit HRAS et pour la logique de Sn pour le chargement des registres à décalage ECL. Pour former HRAS, on fait une conversion ECL, TTL. L'horloge point est une horloge ECL.

HRAS est obtenu avec une bascule $\overline{J}\ \overline{K}$ (ECL) dont les bornes $\overline{J}\ \overline{K}$ sont au niveau 01, dont l'horloge est reliée à $\overline{Sn}$ et dont la borne de forçage à 01 est reliée à fRAS. En effet, pendant que FRAS = 0, on a Q = 0. Quand il retourne au niveau 1, Q est toujours au niveau 0 et le basculement de 0 à 1 de $\overline{Sn}$ remet Q à 1. On fait une conversion ECL,TTL pour avoir HRAS.

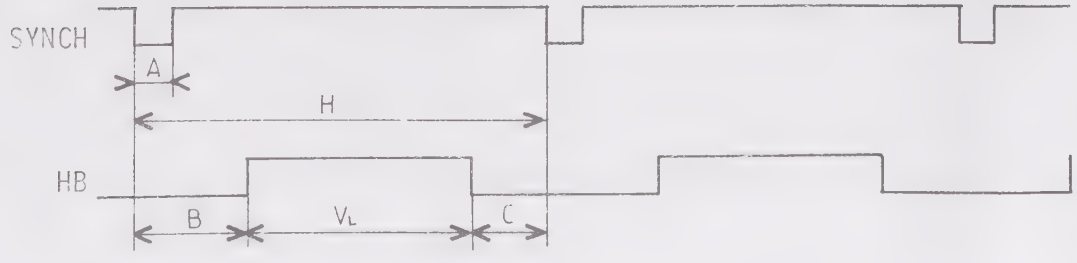


Réalisation des signaux de synchronisation horizontale :

Ils sont imposés par le moniteur télévision et donnés sous forme standard.

Nature des signaux à réaliser : ce sont la synchronisation horizontale et la suppression ligne.

Forme des signaux :

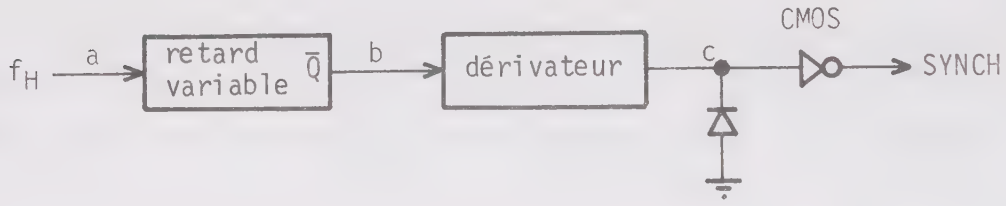


Les valeurs de A,B,C standards en fonction de la résolution sont données par le tableau suivant :

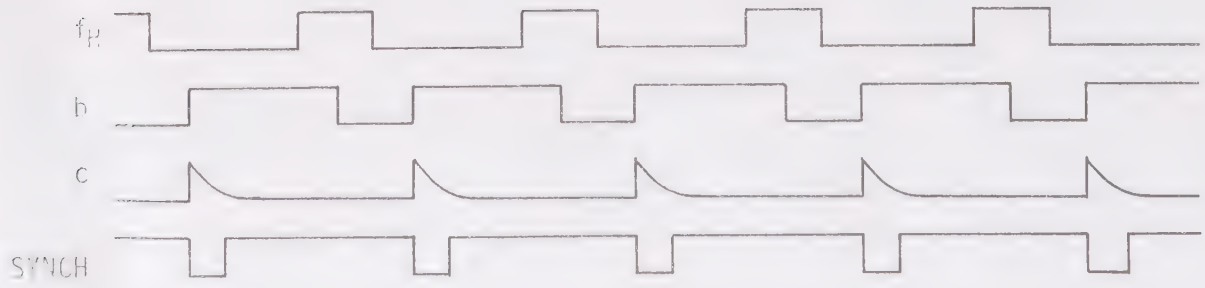
	Haute résolution monochrome 1024 x 1024 non-entrelacé	512 x 512NE 1024 x 1024E	512 x 512E 40-70 Hz	512 x 512E 625 lignes français	640 x 480 525 lignes américain
A	2,8 ± 0,3µs	2,8 ± 0,3µs	4,7 ± 0,5µs	4,8 ± 0,2µs	4,4 à 5,7µs
B	2,93µs	7,5µs MIN	10µs MIN	$11,4\mu s_{MIN}$	$9,4\mu s_{MIN}$
C	0,8µs	0,8µs MIN	1,3µs MIN	$1,4\mu s_{MIN}$	$1,5\mu s_{MIN}$

Synchronisation horizontale SYNCH : elle est obtenue à partir de l'horloge f_H, qui est déphasée afin de permettre un éventuel cadrage de l'image en fonction du déphasage. L'horloge f_H décalée attaque ensuite un dérivateur relié à une porte CMOS pour former les tops lignes dont la durée A est donnée par le tableau ci-dessus.

La porte CMOS sert à éviter les phénomènes transitoires des circuits TTL lors d'un basculement, qui risqueraient de gêner le moniteur télévision.

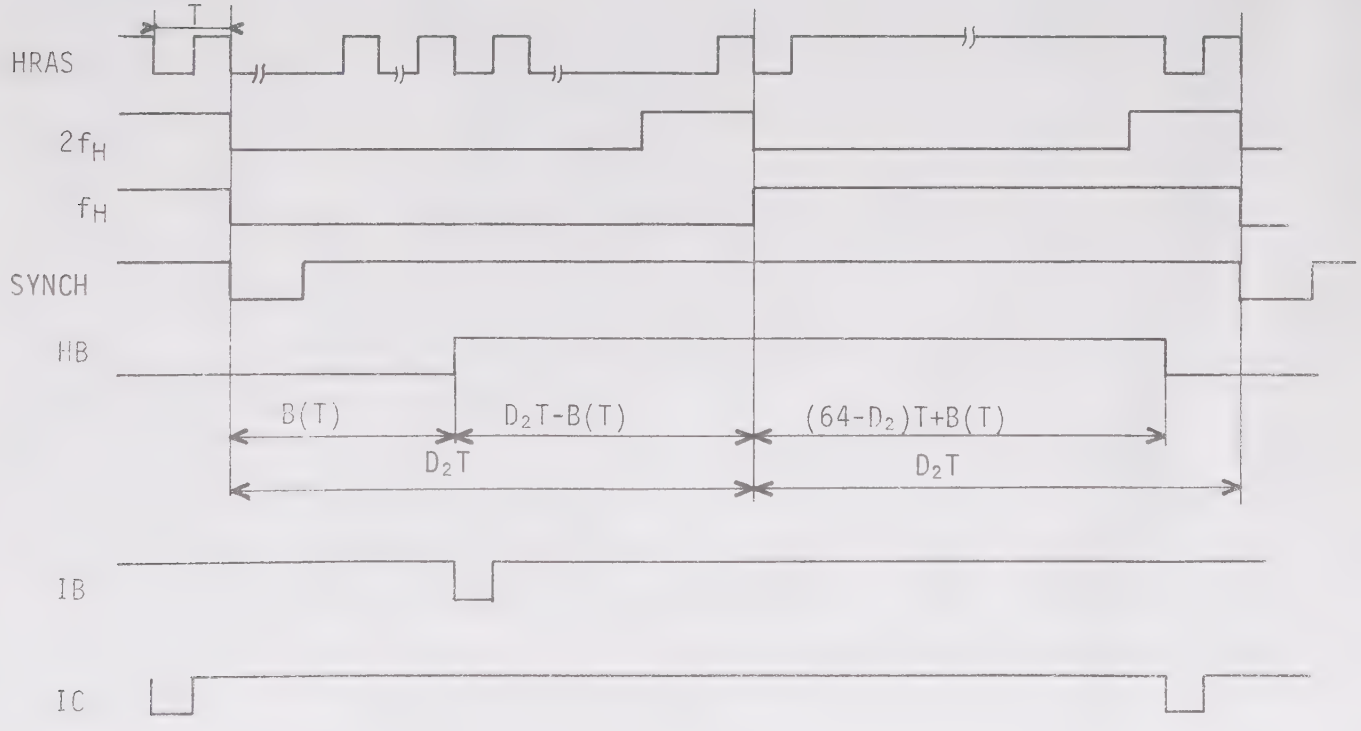


La constante de temps du dérivateur est ajustée suivant la valeur de A.



Suppression ligne HB : la suppression ligne est obtenue à partir de deux impulsions IB et IC obtenues par un décodeur qui décode la valeur des intervalles B et C donnés par le tableau des valeurs standards de B et C, en fonction de la période T de HRAS. On a toujours la durée de visualisation ligne égale à 64T ou 128T. Le tableau ci-dessous donne la valeur de B,C,H en fonction de la période T de HRAS.

	Haute résolution monochrome 1024 x 1024 non-entrelacé	512 x 512NE 1024 x 1024E	512 x 512E 40-70 Hz	512 x 512E 625 lignes français	640 x 480 525 lignes américain
H	86 T	100 T	80 T	160 T	100 T
B	17 T	30 T	14 T	28 T	30 T
C	5 T	6 T	2 T	4 T	6 T



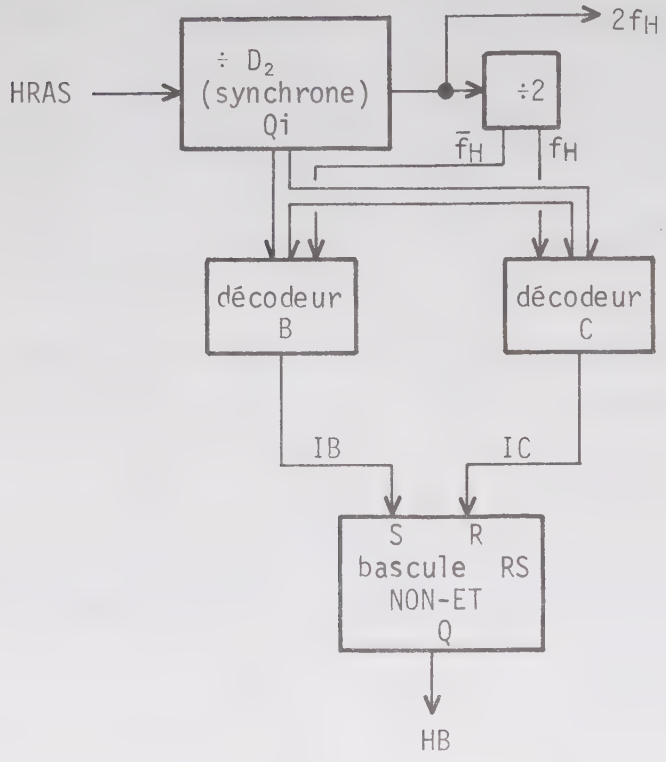
IB ne se produit que quand f_H est à l'état bas à la reconnaissance de B(T)

IC " " f_H " haut " $(64-D_2)T + B(T)$

du diviseur par D_2 de HRAS.

HB est au niveau haut quand IB se produit et à l'état bas quand IC se produit. Il est obtenu par une bascule R S dont R = IC, S = IB et la sortie Q = HB.

Remarque : le fait de décoder les valeurs de B et C entraîne que le diviseur D_2 doit être un compteur synchrone, sinon on aurait des impulsions parasites sur IB et IC.



Nota : si f_H et SYNCH sont en phase, alors l'image réelle sur l'écran est décalée de 7 ou de 11 points par rapport à HB, dû au fait que les registres à décalage ne sont chargés qu'au bout de ces nombres de points par rapport au front descendant de RAS. Mais ce décalage peut être rattrapé en réglant le déphasage de SYNCH par rapport à f_H.

Réalisation des signaux de synchronisation verticale :

Ils sont aussi imposés par le moniteur télévision.

Nature des signaux à réaliser : ce sont : . la synchronisation verticale

. la suppression trame.

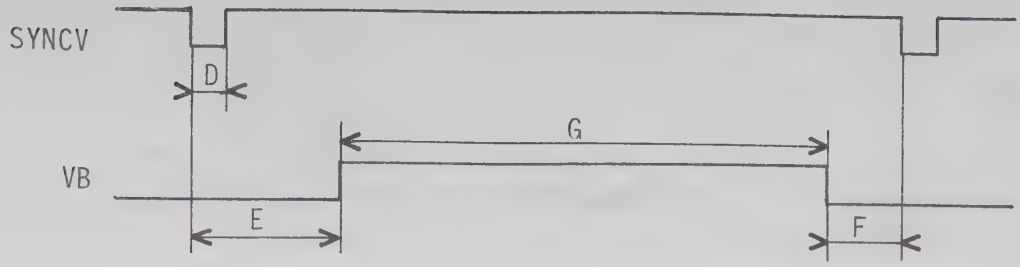


Tableau des valeurs standards des paramètres des signaux de synchronisation verticale.

	1024 x 1024 monochrome non-entrelacé 60 Hz	1024 x 1024 entrelacé	512 x 512NE 50Hz	60Hz	512 x 512E 40-70Hz	512 x 512E 625 lignes	640 x 480 525 lignes
D	6 H	3 H	5 H	6 H	3 H	2,5 H	3 H
E	20 H Min	20 H Min	20 H Min	id	15 H Min	20 H Min	17 H Min
F	7 H Min	7 H Min	7 H Min	id	3 H Min	2,5 H	3 H
G	1024 H	512 H	512 H	id	256 H	256 H	240 H

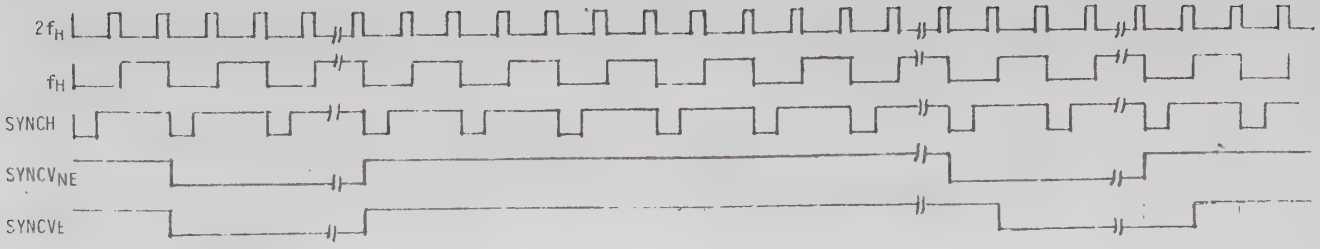
Synchronisation verticale SYNCVE ou SYNCVNE :

1ère solution uniquement en mode non-entrelacé : SYNCVNE est obtenu par le même procédé que SYNCH à partir de fv.

2ème solution : par l'utilisation du diviseur par le nombre total de lignes D_3. C'est le compteur demi-ligne si l'horloge est $2f_H$ et le compteur ligne si l'horloge est f_H.

En mode entrelacé, l'allure de SYNCV est un peut différente. Un des tops de SYNCVE pour l'une ou l'autre trame doit commencer au milieu d'une ligne, ce qui nécessite l'utilisation de la fréquence demi-ligne $2f_H$.

Allure de SYNCV dans les 2 modes :



La première solution est laissée au profit de la deuxième qui permet de passer d'un mode à un autre en changeant la valeur du compteur demi-ligne.

On prend comme origine des temps, le temps où le compteur demi-ligne est à zéro. A chaque passage du compteur demi-ligne à la valeur 6 (correspondant à 3 H) ou (5 correspondant à 2,5 H), SYNCV est mis au niveau bas et au passage de la valeur 12 (intervalle 3 H à partir de 6) ou 10, SYNCV retourne au niveau haut. Suivant l'horloge du diviseur D_3, $2f_H$ ou f_H, on est dans le mode entrelacé ou non-entrelacé.

SYNCV est réalisé avec une bascule RS dont les entrées RS sont des impulsions f_6 f_{12} délivrées par des décodeurs par 6 et par 12 de la valeur du compteur demi-ligne.

On a donc :

$R = f_{12}$
$S = f_6$
$SYNCV = \overline{Q}$

le Reset doit être du côté de R.

Remarque : seuls les quatre premiers étages du compteur demi-ligne doivent être synchrones.

Suppression trame VB : elle est obtenue par le même procédé que pour l'obtention de SYNCV. Seules les valeurs du compteur ligne ou demi-ligne à décoder changent. Elles sont données par le tableau ci-dessous pour les différentes résolutions.

	1024 x 1024NE	1024 x 1024E	512 x 512NE	512 x 512E 50Hz 60Hz	512 x 512E 625 lignes	640 x 480 525 lignes
f_1	31 H	74 $\frac{H}{2}$	37 H	74 $\frac{H}{2}$ 52 $\frac{H}{2}$	74 $\frac{H}{2}$	36 $\frac{H}{2}$
f_2	31 + 1024 H	74 + 1024 $\frac{H}{2}$	37 + 512 H	74 + 512 $\frac{H}{2}$ 52 + 512 $\frac{H}{2}$	74 + 512 $\frac{H}{2}$	36 + 480 $\frac{H}{2}$

Parité trame PT uniquement au mode entrelacé : elle est obtenue par la division par deux de la fréquence verticale, c-à-d la dernière sortie du compteur ligne ou demi-ligne. La remise à zéro de la bascule se fait en même temps que celle de tous les diviseurs.

Réalisation du signal de synchronisation mélangé : pour certains types de moniteur télévision, on a besoin de mélanger la synchronisation horizontale et verticale. Dans ce cas, pendant l'intervalle de top trame, le générateur de balayage horizontal du moniteur télévision ne recevant plus aucune impulsion de synchronisation va osciller en régime libre, ce qui entraîne en début de trame que le spot pourrait se trouver dans un endroit quelconque de la première ligne. C'est ce qui a amené les standards à découper le top trame au rythme du cycle ligne. C'est la serration du top trame.

Cas du mode entrelacé : pour que les trames paire et impaire aient la même durée, les standards ont encadré le top trame par deux impulsions d'égalisation de période $\frac{H}{2}$.

Enveloppe d'égalisation serration e : dans le cas où toutes les lignes sont utilisées, c'est le signal de suppression trame e = VB. Sinon elle est obtenue de la façon suivante : à chaque passage à zéro du compteur demi-ligne, elle doit être mise au niveau bas. Dès la reconnaissance de 18 ou 15, correspondant à 9 H ou 7,5 H suivant le standard utilisé, elle doit être au niveau haut.

Le passage à zéro du compteur demi-ligne correspond au front descendant de la fréquence verticale c-à-d sa dernière sortie. Donc, quand fv passe de 1 à 0, e = 0 et, dès qu'on décode la valeur 18 ou 15, e = 1. L'enveloppe est réalisée avec une bascule JK dont les entrées J et K sont au niveau haut ; l'horloge est reliée à fv et la borne de forçage à 1 est reliée au décodeur de la valeur 18 ou 15.

Signal d'égalisation Eg : il est obtenu de la même manière que SYNCH mais avec l'horloge $2f_H$ et sans déphasage.

Signal de serration Ser : il est obtenu de la même manière que Eg mais avec une largeur d'impulsion plus grande.

Signal de synchronisation mélangé SYNC : . en mode entrelacé,

SYNC est égal à SYNCH quand VB = 1 et e = 1

" Ser " VB = 0 et e = 0

" Eg " VB = 1 et e = 0

Ce qui donne : $\boxed{SYNCE = \overline{SYNCV}.Ser + e.SYNCH + Eg.\ \overline{e}\ SYNCV}$

. en mode non-entrelacé

SYNC est égal à SYNCH quand SYNCV = 1

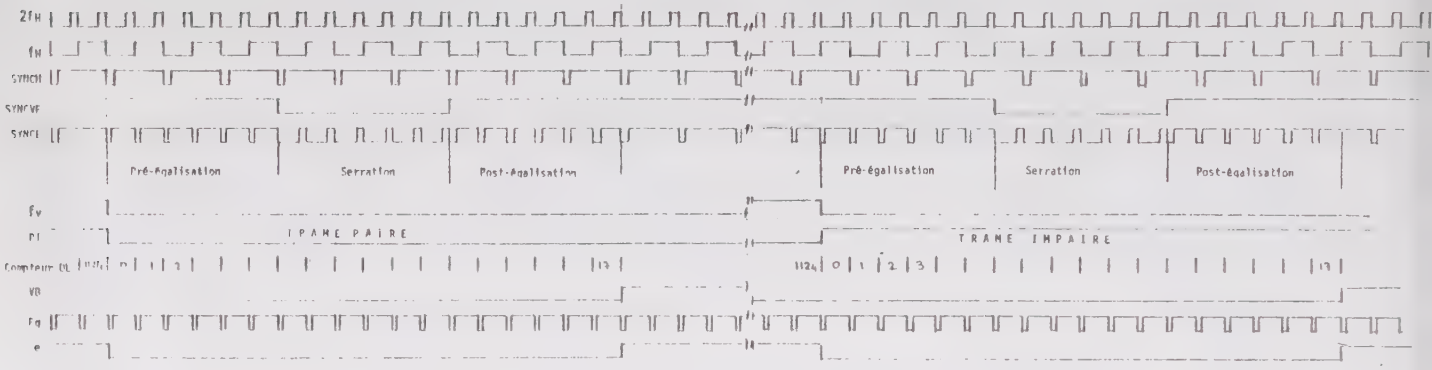
" Ser " SYNCV = 0

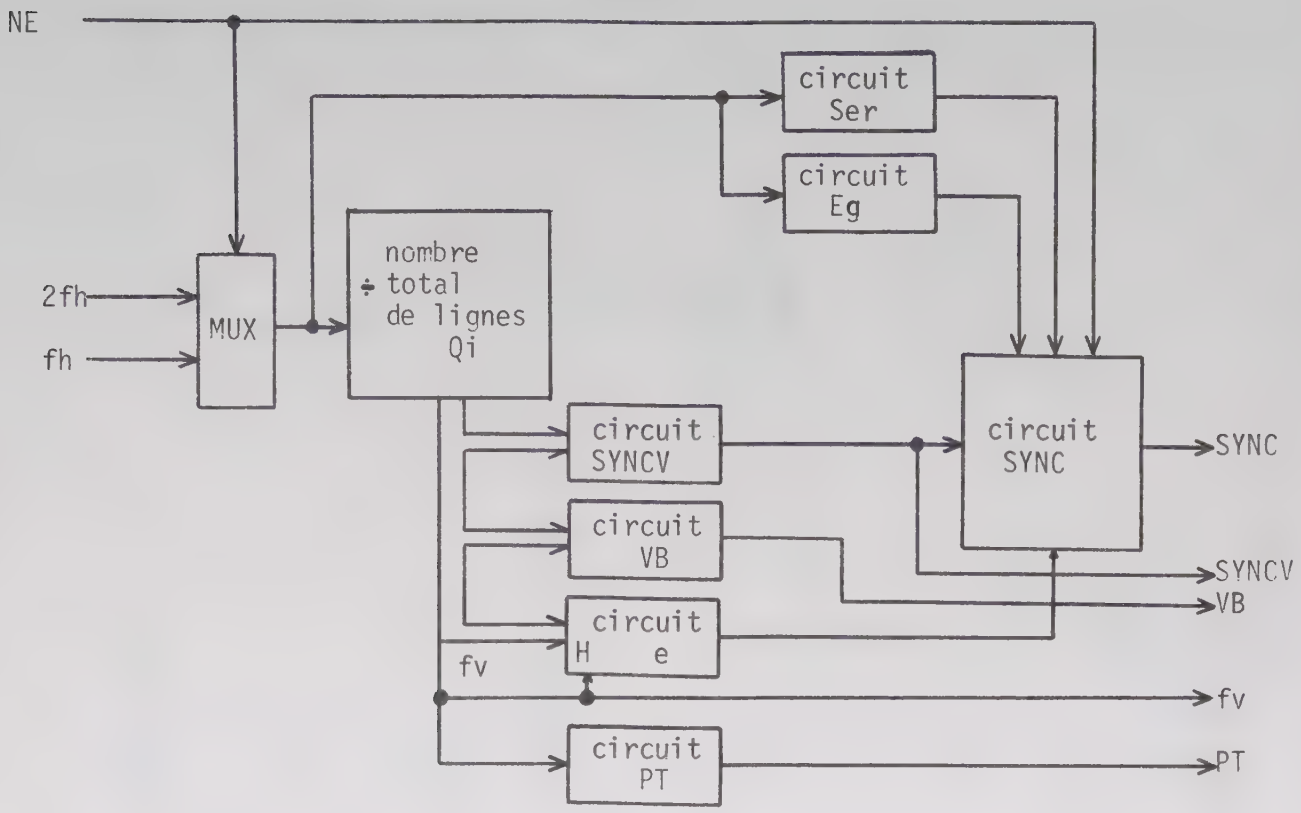
Ce qui donne : $\boxed{SYNCNE = SYNCV.SYNCH + \overline{SYNCH}.Ser}$

En tout, SYNC est le multiplexage de SYNCE et de SYNCNE suivant NE.

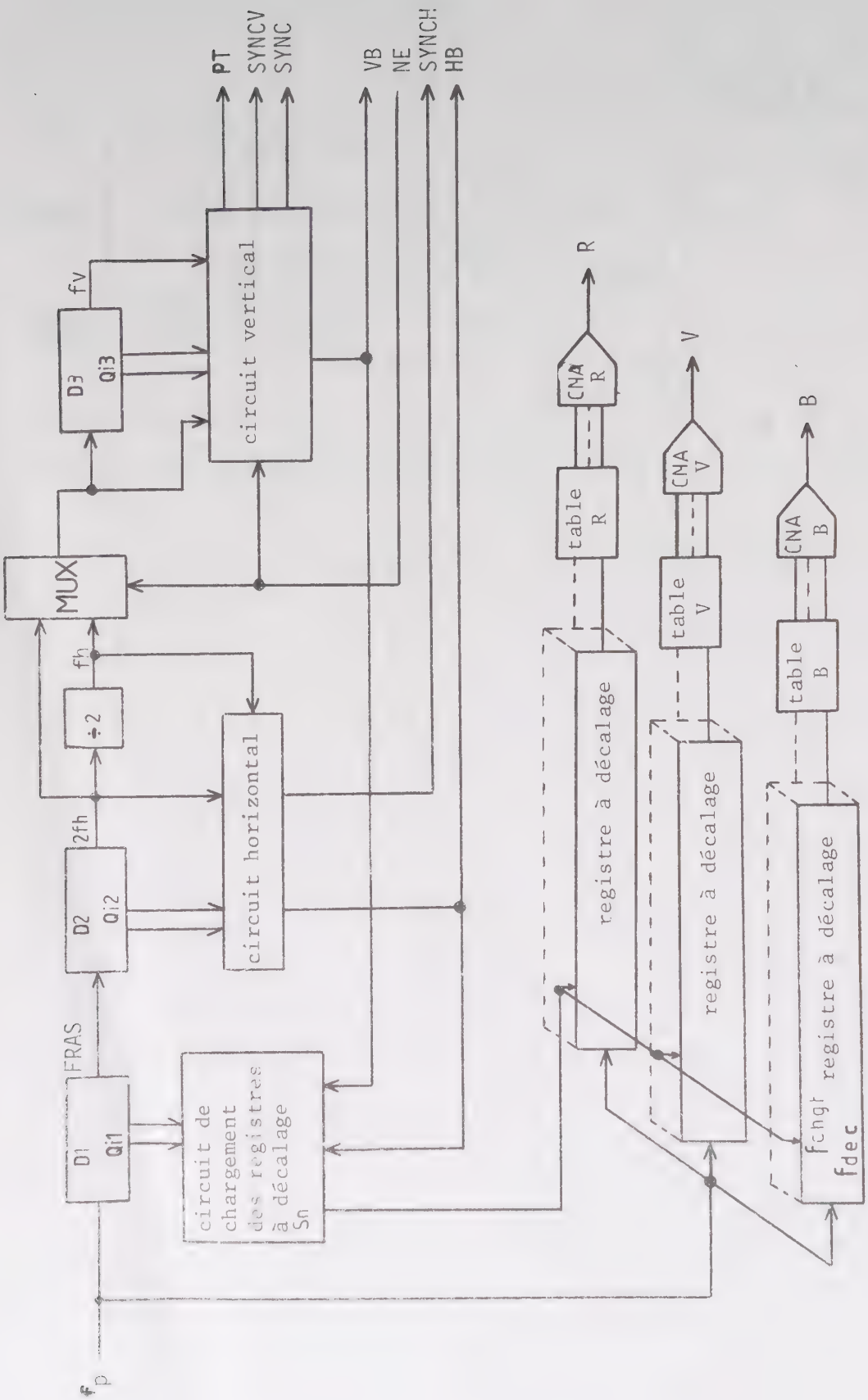
$$\boxed{SYNC = \overline{NE}.SYNCNE + NE.SYNCE}$$

Allure des signaux en mode entrelacé :





ORGANISATION DU CIRCUIT DE SYNCHRONISATION VERTICALE



ORGANISATION DE L'ORGANE VIDEO

CHAPITRE V

CONCLUSION

CONCLUSION

Cette étude se situe dans le cadre général d'un système graphique de synthèse d'image :

- d'un côté a été étudié une architecture multiprocesseurs,

- d'un autre côté, une unité d'affichage connectable à ce système multiprocesseur.

Une réalisation des différents modules (en partie simulée) tenant compte des contraintes matérielles (volume des composants) nous a permis d'apprécier la valeur du système proposé. A l'heure actuelle, elle n'est certes pas satisfaisante mais mène à des prolongements évidents :

- étude de l'intégration des différents modules de l'unité d'affichage et de l'ensemble multiprocesseurs ;

- faire fonctionner l'ensemble multiprocesseur plus l'unité d'affichage ;

- rechercher une nouvelle forme d'algorithme parallèle implémentable sur cette architecture, éventuellement pour des types d'objets spécifiques.

CHAPITRE VI

BIBLIOGRAHIE

(Agra 82) D.P. AGRAWAL
Testing and Fault Tolerance of Multistage Interconnection Networks
Computer Vol.15 Apr.82

(BeBa 80) A. BECHTOLSHEIM - F. BASKETT
High-Performance Raster Graphics for Microcomputer Systems
SIGGRAPH'80 Conference Proceedings Vol.14,n°3 Jul.80

(BeCaGi) R. BEAUFILS - R. CAUBET - A. GISSELBRECHT
Etude des Compromis Matériel-Logiciel dans un Système d'Affichage d'Images sur un Ecran Vidéo en vue de Faciliter la C.A.O
Laboratoire Langages et Systèmes Informatiques
Université Paul Sabatier Toulouse

(Baec 79) R. BAECKER
Didital Video Display Systems and Dynamic Graphics
SIGGRAPH'79 Proceedings Vol.13 n°2 Aug.79

(CrHo 81) F.C. CROW - M.W. HOWARD
A Frame Buffer System with Enhanced Functionality
SIGGRAPH'81 Proceedings Vol.15 n°3 Aug.81

(ChJa 81) P. CHANG - R. JAIN
A Multi-Processor System for Hidden-Surface-Removal
Computer Graphics Vol.15 n°4 Dec.81

(Crow 78) F.C. CROW
The Use of Grayscale for Improved Raster Display of Vectors and Characters
SIGGRAPH'78 Proceedings Vol.12 n°3 Aug.78

(Conr 80) CONRAC Division
Raster Graphics Handbook

(CHPCH 79) C. CSURI - R. HACKATHORN - R. PARENT - W. CARLSON - M. HOWARD
Towards an Interactive High Visual Complexity Animation System
SIGGRAPH'79 Proceedings Vol.13 n°2 Aug.79

(Davi 80) L.S. DAVIS
Computer Architectures for Image Processing
P oceedings of the Workshop on Picture Data Description and Management
Aug.80

(DiJu 81) D.D. DIAS - J.R. JUMP
Analysis and Simulation of Buffered Delta Networks
IEEE Transaction on Computers Vol.C30 n°4 Apr.81

(Dill 81) J.C. DILL
An Application of Color Graphics to the Display of Surface Curvature
SIGGRAPH'81 Proceedings Vol.15 n°3 Aug.81

(Duff 79) T. DUFF
Smoothly Shaded Rendering of Polyhedral Objects on Raster Displays
SIGGRAPH'79 Proceedings Vol.13 n°2 Aug.79

(Efci) EFCIS
Processeur de Visualisation Graphique
Manuel d'Utilisation

(Ferr 81) M. FERRETTI
Des Ecrans Couleurs Interactifs à Haute Résolution pour les Terminaux de Demain
Sciences et Techniques n°79-80 Aoû.81

(Fole 79) J. FOLEY
Some Raster Graphics Extentions to the Core System
SIGGRAPH'79 Proceedings Vol.13 n°2 Aug.79

(Fran 81) M.A. FRANKLIN
VLSI Performance Comparison of Banyan and Crossbar Communication Networks
IEEE Transaction on Computers Vol.C30 n°4 Apr.81

(Fuch 79) H. FUCHS
An Expandable Multiprocessor Architecture for Video Graphics
Proceedings 6th. Symposium on Computer Architecture Apr.79

(Fuch 79) H. FUCHS
Generating Smooth 2D Monocolor Line Drawing on Video Displays
SIGGRAPH'79 Proceedings Vol.13 n°2 Aug.79

(Gard) T.N. GARDNER
Raster Display on a Line Drawing Graphics System
Adage Inc.
One Fortuna Drive, Billerica, Massachussetts 01821

(GrHe 81) R.D. GRAPPEL - J.E. HEMENWAY
Color-Graphics Techniques Say it with Pictures
EDN May 81

(Grav 80) M. GRAVE
Etude d'un Noyau de Système de Synthèse d'Image,Application à la Visualisation de Scènes Tridimensionnelles
Thèse de Docteur-Ingénieur,Université des Sciences et Techniques de Lille,Décembre 1980

(GSS 81) S. GUPTA - R.F. SPROULL - I.E. SUTHERLAND
A VLSI Architecture for Updating Raster Scan Displays
SIGGRAPH'81 Proceedings Vol.15 n°3 Aug.81

(JoGr 78) G.H. JOBLOVE - D. GREENBERG
Color Spaces for Computer Graphics
SIGGRAPH'78 Proceedings Vol.12 n°3 Aug.78

(KaSuCh 75) J.T. KAJIYA - I.E. SUTHERLAND - E.C. CHEADLE
A Random Access Video Frame Buffer
Proceedings of the Conference on Computer Graphics,Pattern Recognition and Data Structures
IEEE Catalog n°75CH09181-IC May 1975

(KaGr 79) M. KAPLAN - D.P. GRENBERG
Parallel Processing Techniques for Hidden Surface Removal
SIGGRAPH'79 Proceedings Vol.13 n°2 Aug.79

(Krum) L.L. KRUMMEL
Color Display Hardware
Ramtek Corporation , California

(LeNe 81) L. LEVINSON - E. NEWBERT
Microprocessor-Controlled System prints Color TV Pictures
Electronics Sep.81

(LoCu 80) R.M. LOUGHEED - D.L. Mc CUBBREY
The Cytocomputer: A Practical Pipelined Image Processor
Proceedings of the Workshop on Picture Data Description and Management Aug.80

(Math 78) P. MATHERAT
Conception d'un Circuit Intégré pour la Visualisation Graphique
Thèse de 3ème Cycle , Institut de Programmation de Paris
Mai 1978

(Méri 79) M. MERIAUX
Etude et Réalisation d'un Terminal Graphique Couleur Tridimensionnel fonctionnant par Taches
Thèse de Docteur-Ingénieur , Université des Sciences et Techniques de Lille , Janvier 1979

(Mich 77) G.L. MICHEL
Un Processeur Spécialisé pour les Applications Graphiques et le Traitement du Signal : PEGASE
Thèse de Docteur-Ingénieur , Université de Paris XI , Juin 1977

(Mors 79) A. MORSE
Some Principles for the Effective Display of Data
SIGGRAPH'79 Proceedings Vol.13 n° 2 Aug.79

(Nata 81) J.A.G. NATAREN
M.G.I.T.: Un Modeleur Géométrique Interactif Tridimensionnel
Thèse de Docteur-Ingénieur présentée à 1' Ecole Nationale Supérieure de Techniques Avancées , Juin 1981

(Nec) NEC
µPD 7220/GDC , Graphics Display Controller
Manuel d'utilisation NEC

(NeSp 79) W.M. NEWMAN - R.F. SPROULL
Principles of Interactive Computer Graphics
Mc GrawHill 2nd Edition 1979

(Park 79) F.I. PARKE
Simulation and Expected Performance Analysis of Multiple Processor Z-Buffer Systems
SIGGRAPH'79 Proceedings Vol.13 n°2 Aug.79

(PaFu 79) B. PARVIN - K.S. FU
A Microprogrammable Vector Processor for Image Processing Applications
Technical Report , School of E.E. , Purdue University Dec.79

(Pate 81) J.H. PATEL
Performance of Processor-Memory Interconnections for Multiprocessors
IEEE Transactions on Computers Vol.C30 n°10 Oct.81

(PeLa 81) PEN-CHUNG YEW - D.H. LAWRIE
An Easily Controlled Network for Frequently Used Permutations
IEEE Transactions on Computers Vol.C30 n°4 Apr.81

(PiBe) R.J. PISTENON - C. BELLEVAUX
L'Espace 3D et la CAO
Centre CPAO ENSTA

(RaRö 80) J. RAKOTOMALALA - H. RÖTHLISBERGER
F.I.R.S.T : A High Resolution Raster Scan Display System for Interactive Layout of Text and Figures
Proceedings of the International Conference and Exhibition
University of Genova , Switzerland Sep.80

(Shou 79) R.G. SHOUP
Color Table Animation
SIGGRAPH'79 Proceedings Vol.13 n°2 Aug.79

(SiMi 81) H.J. SIEGEL - R.J. Mc MILLEN
The Multistage Cube : a Versatile Interconnection Network
Computer Vol.14 n°12 Dec.81

SSKMSD 81) H.J. SIEGEL - L.J. SIEGEL - F.C. KEMMERER - P.T. MUELLER - H.E. SMALLEY - S.D. SMITH
PASM : a Partitionnable SIMD/MIMD System for Image Processing and Pattern Recognition
IEEE Transactions on Computers Vol.C80 n°12 Dec.81

(Smad 79) M.D. SMADJA
Etude d'un Système de Construction de Volume
Thèse de 3ème Cycle , Université de Paris XI Novembre 1979

(Smit 78) A.R. SMITH
Color Gamut Transform Pair
SIGGRAPH'78 Proceedings Vol.12 n°3 Aug.78

(Sper 77) SPERRY UNIVAC
Color Terminal Developpement Study
Final Technical Report Aug.77

(Tani 80) S.L. TANIMOTO
Color-Mapping Techniques for Computer-Aided Design and Verification of VLSI System
Computer and Graphics Vol.5 n°2-4 1980

(TDF) TELEDIFFUSION DE FRANCE
Dossiers Techniques de Radiodiffusion et Télévision
Télédiffusion de France

(TI) TEXAS INSTRUMENTS
TMS 9918 Video Display Processor , Data Manual
Texas Instruments

(TLE 81) TOUTE L'ELECTRONIQUE
Covertisseurs N/A Pour Ecrans Graphiques
Toute l'Electronique n°468 Nov.81

(Tsé 81) TSE-YUN FENG
A Survey of Interconnection Networks
Computer Vol.14 n°12 Déc.81

(Vill 79) A. VILLAVICENCIO RAMIREZ
Conception et Réalisation d'un Système Graphique Interactif autour d'une Architecture à Microprocesseurs Multiples : Application dans le Contexte CORAIL
Thèse de Docteur-Ingénieur , Institut National Polytechnique de Grenoble , Septembre 1979

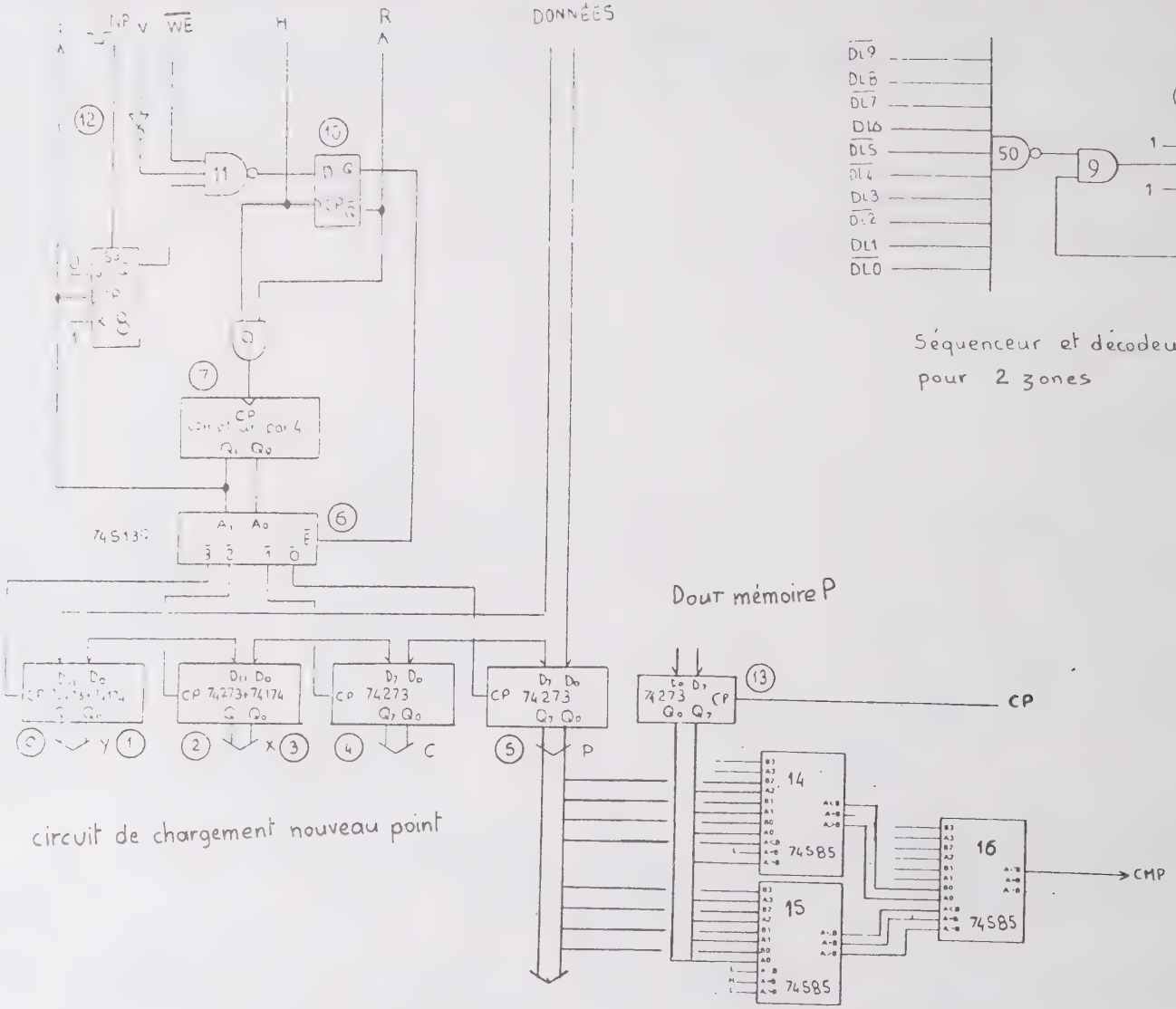
(Wala 80) A.M. WALABY
Fast Colour Raster Graphics using an Array Processor
Proceedings of the International Conference and Exhibition
University of Geneva Switzerland Sep.80

(Wein 81) R. WEINBREG
Parallel Processing Image Synthesis and Anti-Aliasing
SIGGRAPH'81 Vol.15 n°3 Aug.81

CHAPITRE VII

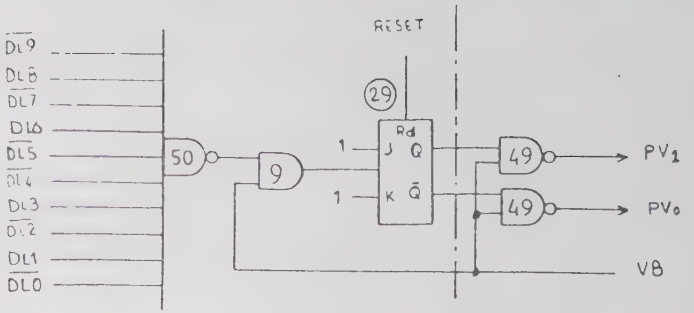
ANNEXES :

CONFIGURATION DE TESTS



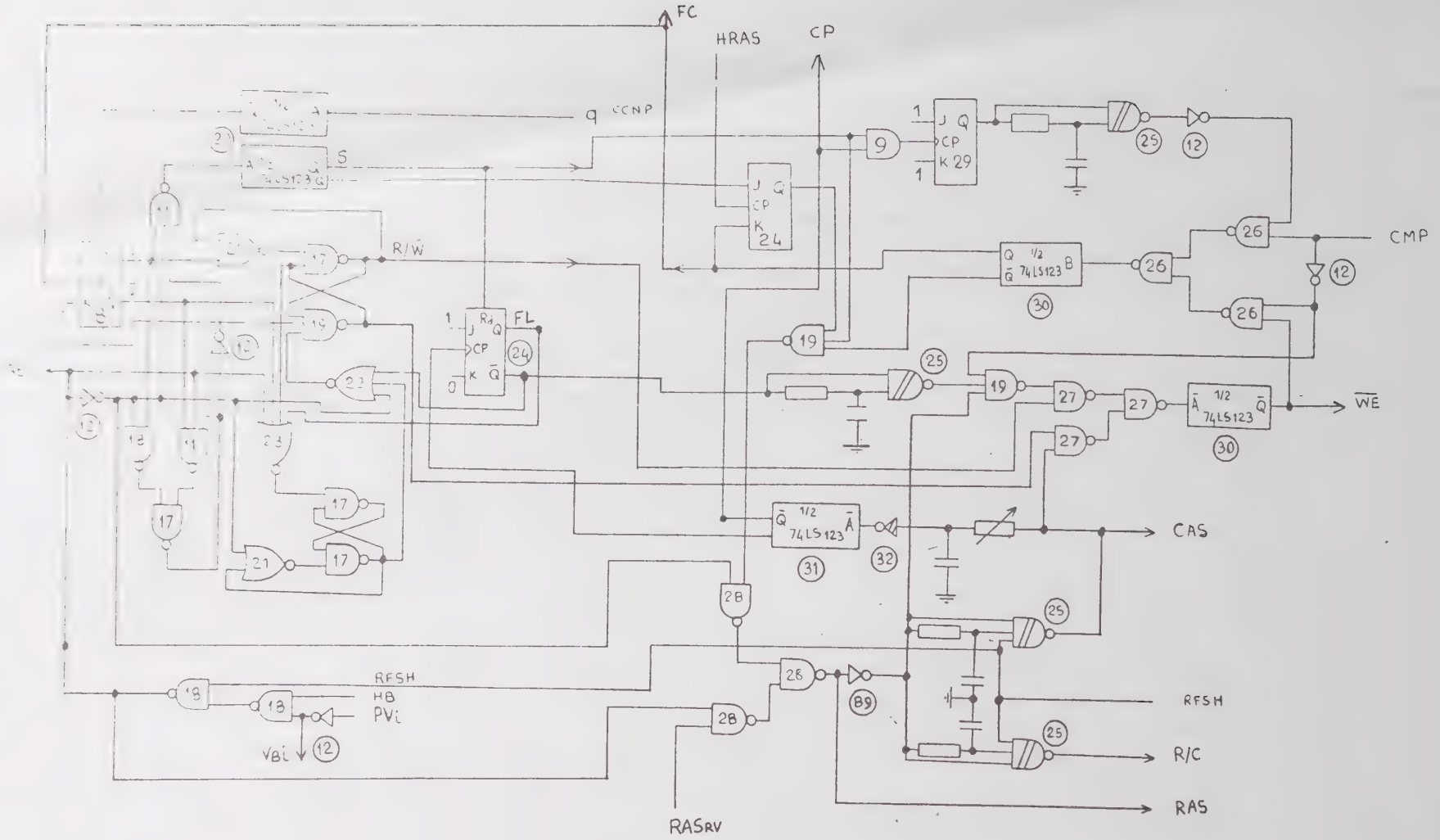
circuit de chargement nouveau point

comparateur



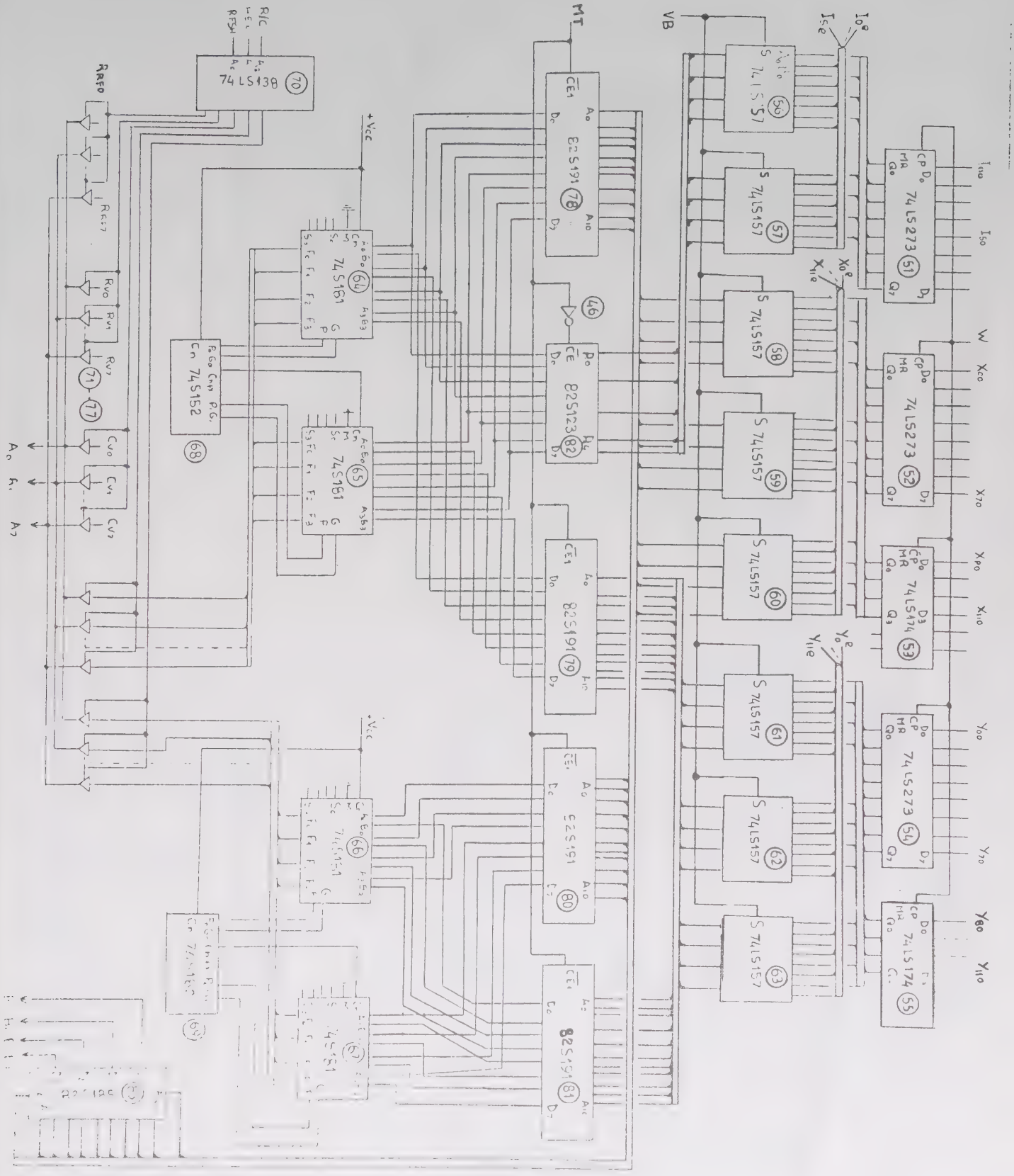
Séquenceur et décodeur de la zone visualisée pour 2 zones

CIRCUIT DE MISE A JOUR DE L'IMAGE



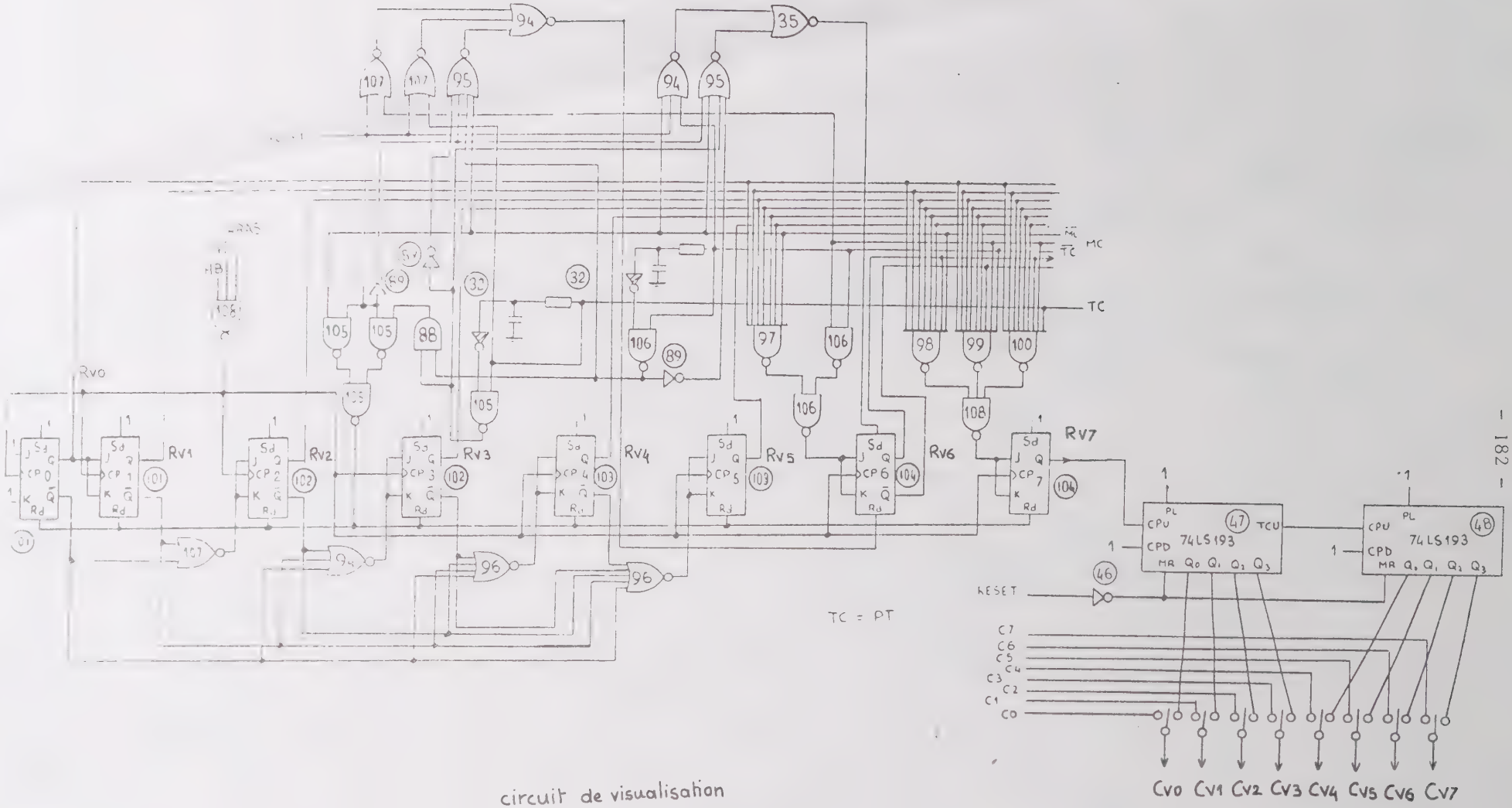
circuit de lecture et de chargement de l'ancien point

CIRCUIT DE MISE A JOUR DE L'IMAGE



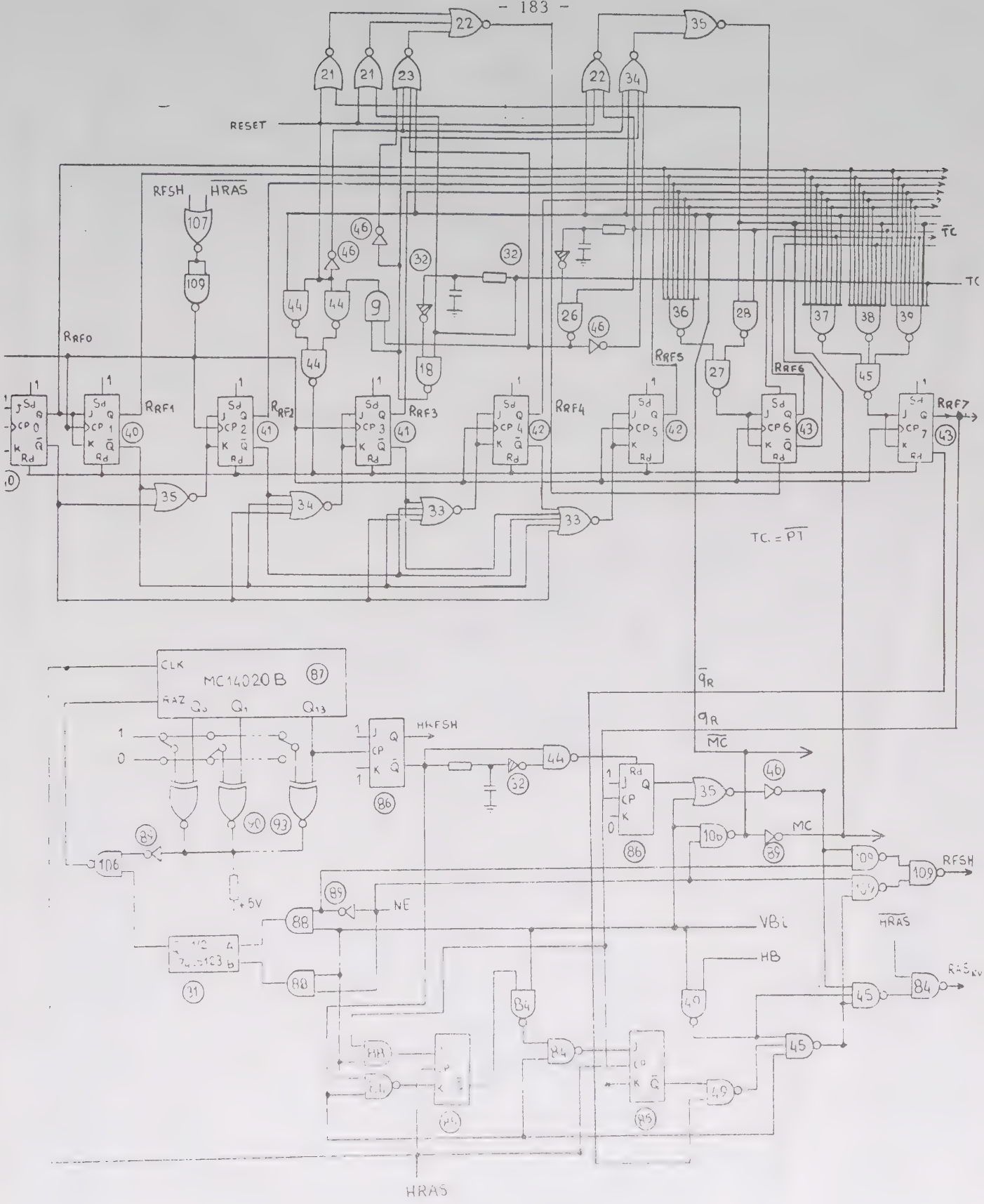
circuit d'adressage

CIRCUIT DE MISE A JOUR DE L'IMAGE



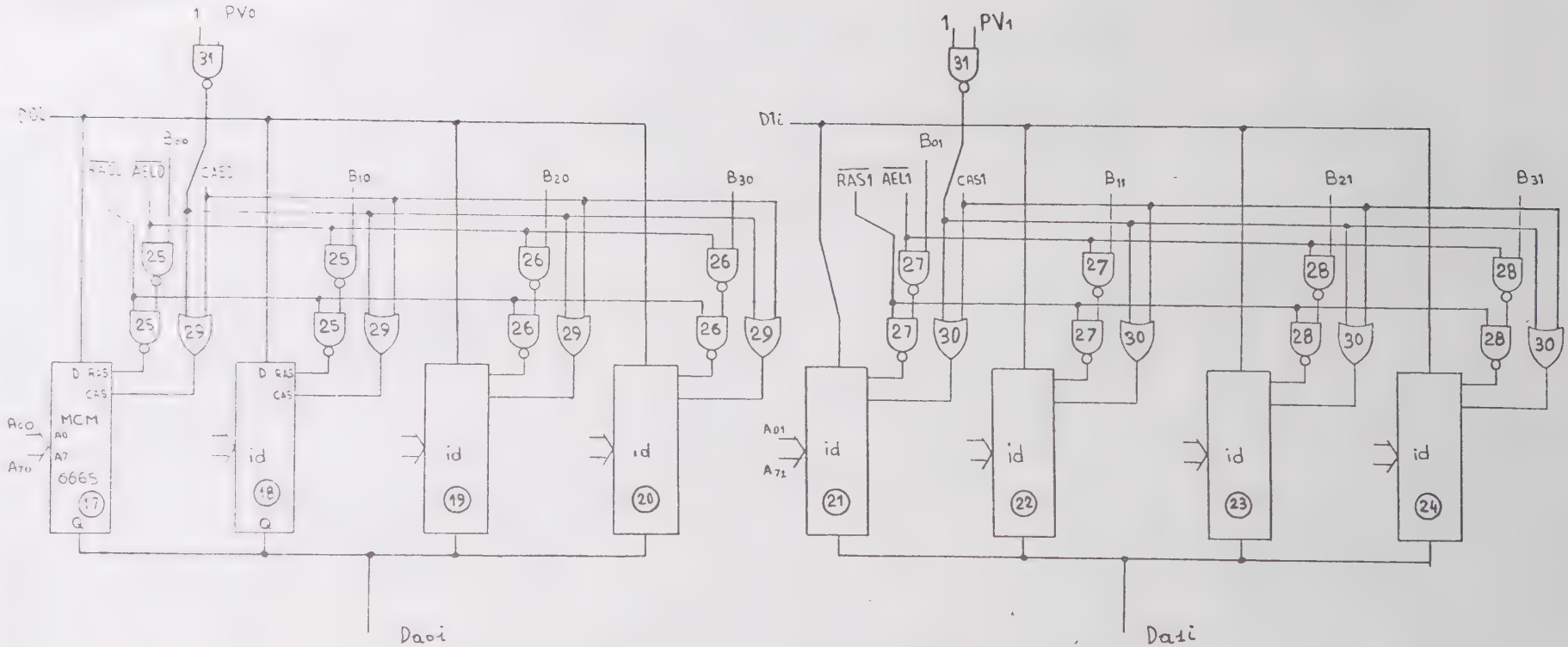
circuit de visualisation

CIRCUIT DE MISE A JOUR DE L'IMAGE



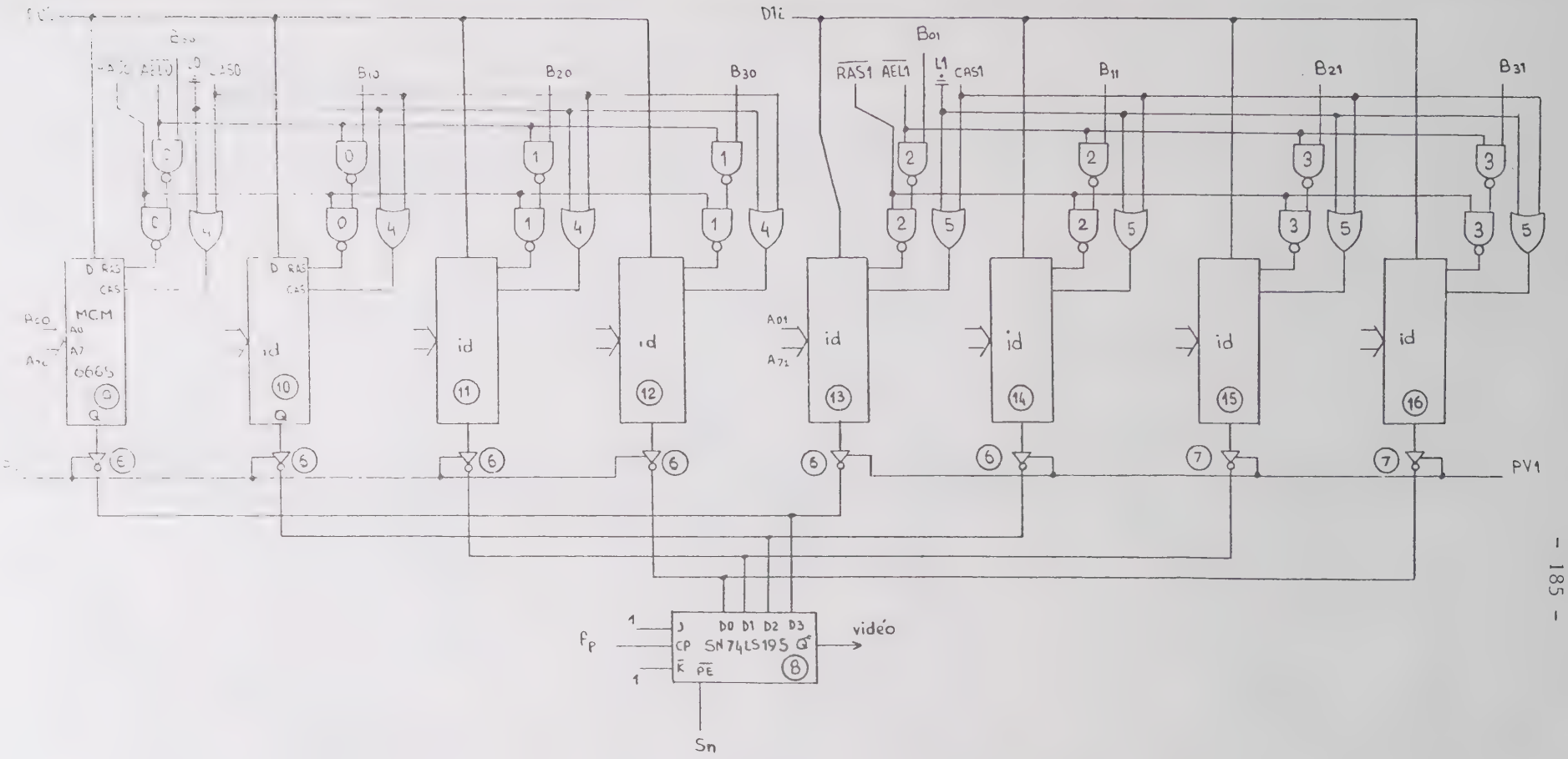
circuit de rafraichissement

CIRCUIT DE MISE A JOUR DE L'IMAGE



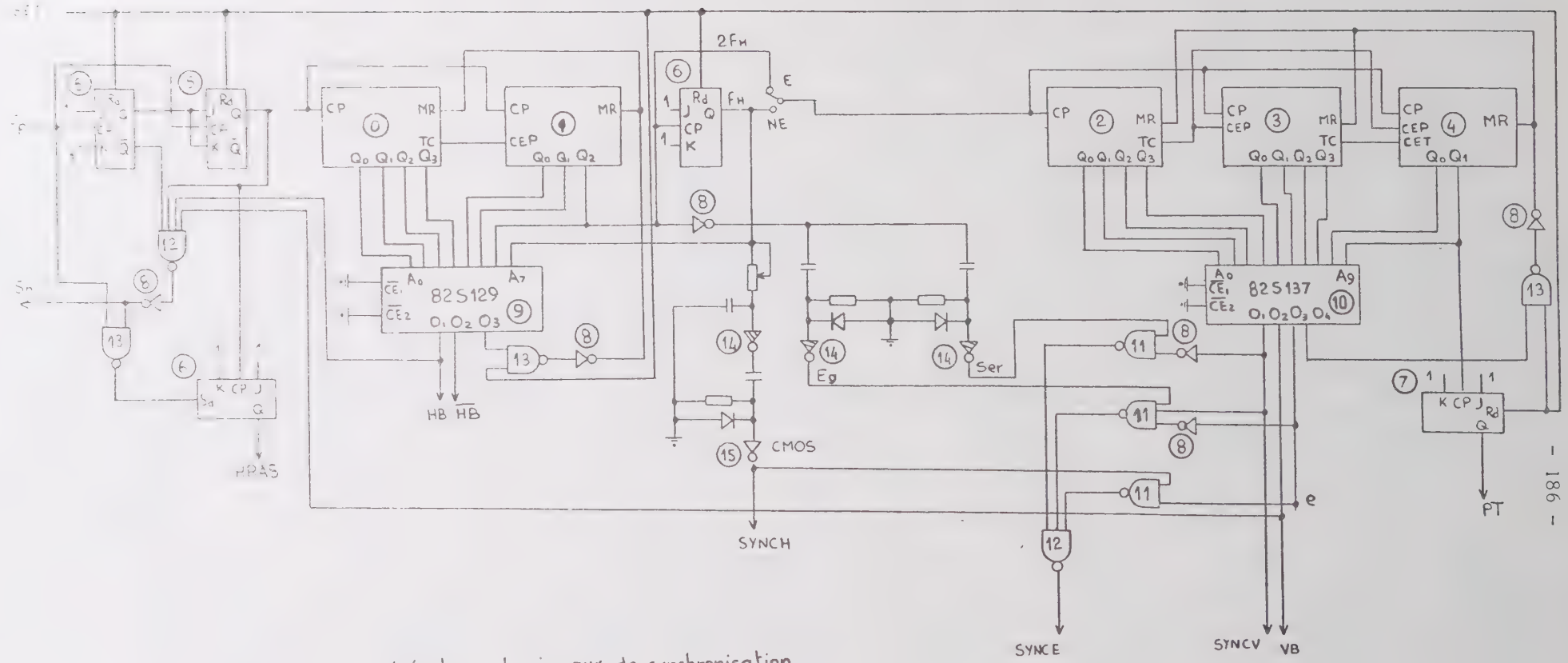
plan profondeur

PLAN MÉMOIRE D'IMAGE



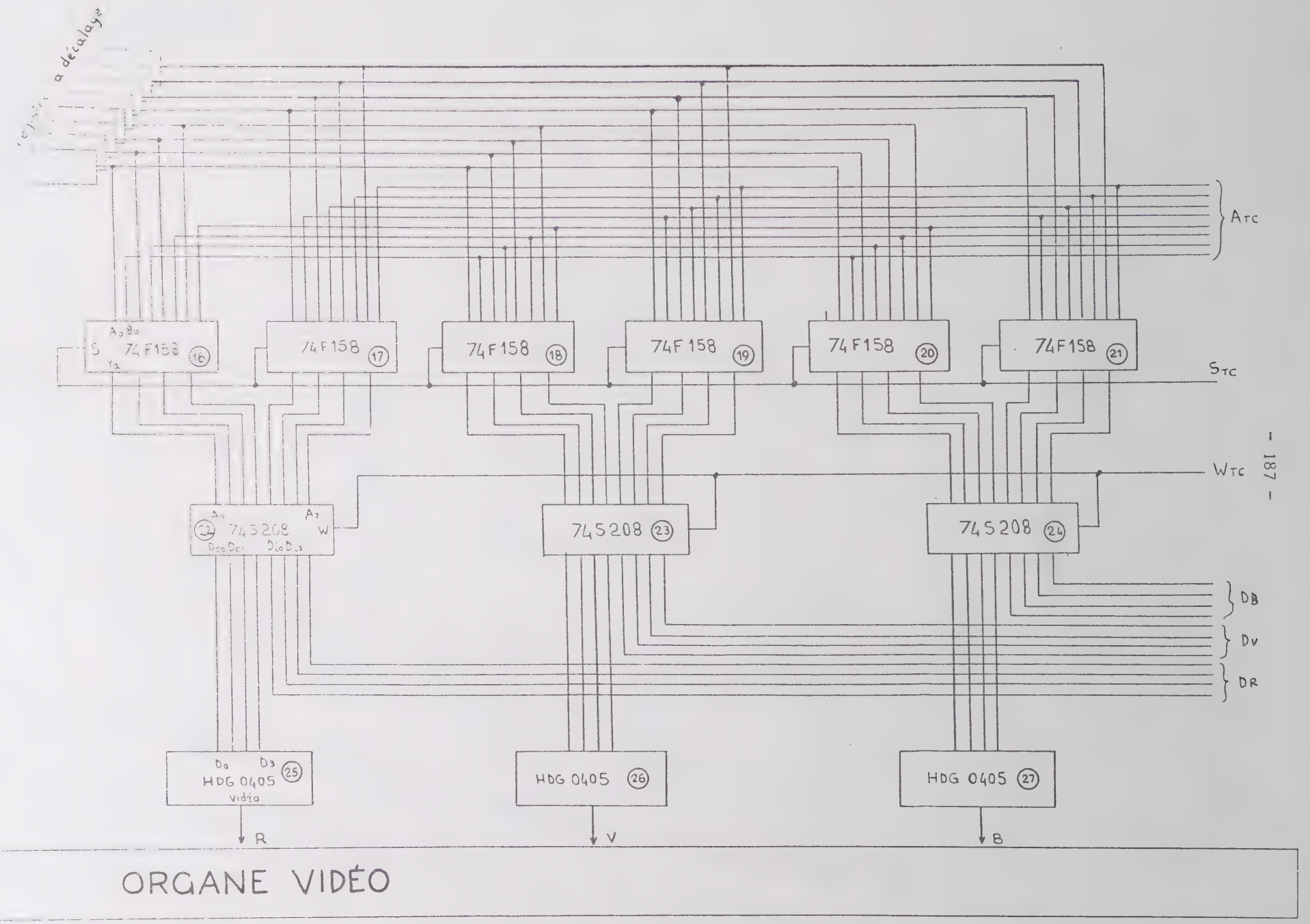
plan couleur

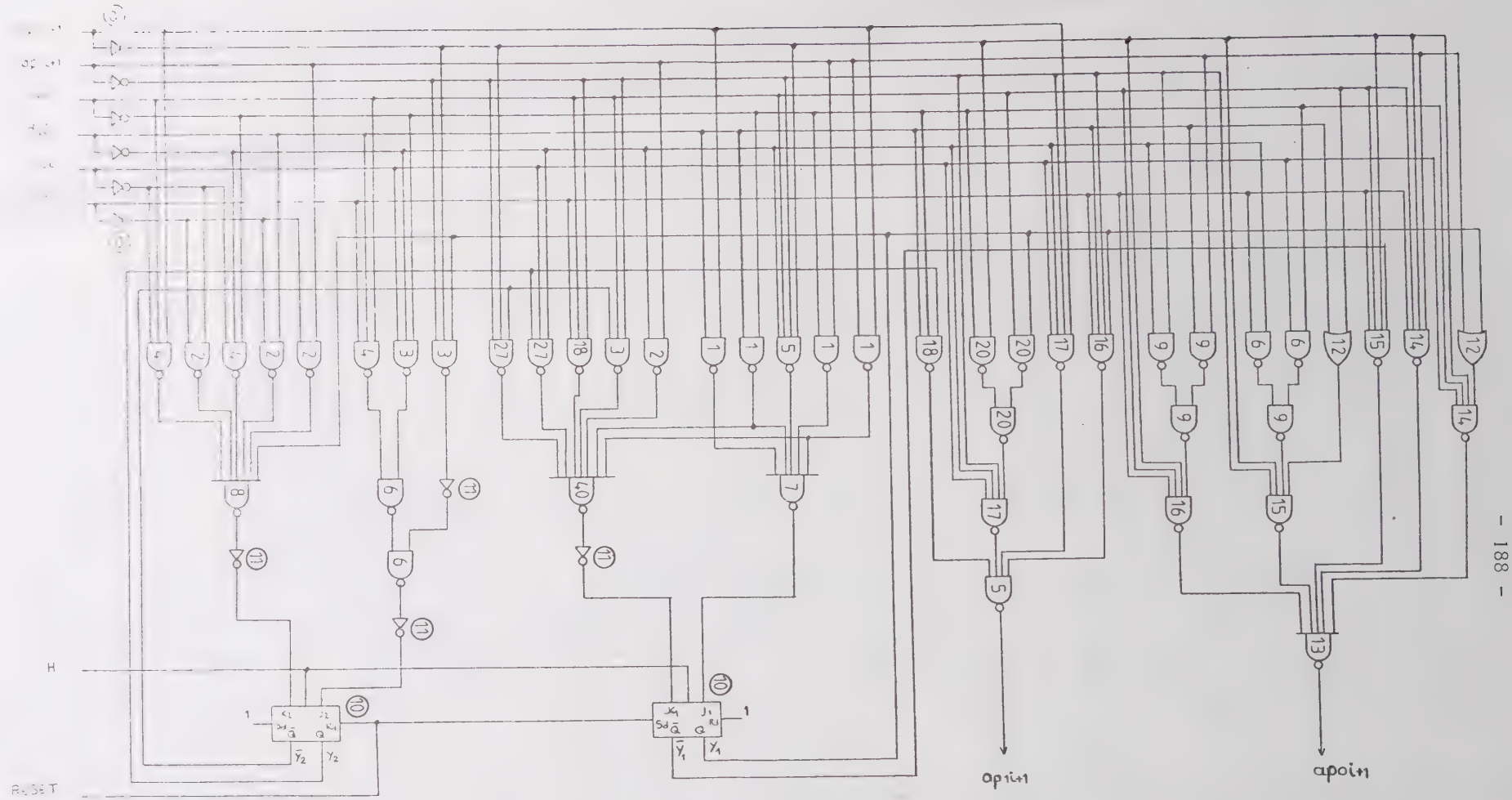
PLAN MÉMOIRE D'IMAGE



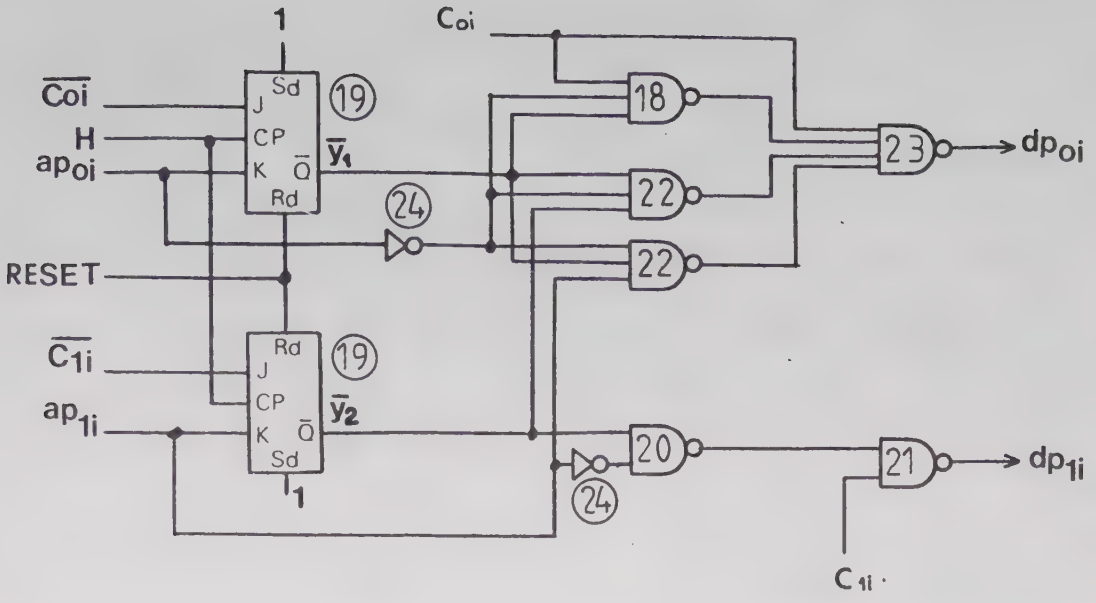
générateur de signaux de synchronisation

ORGANE VIDÉO

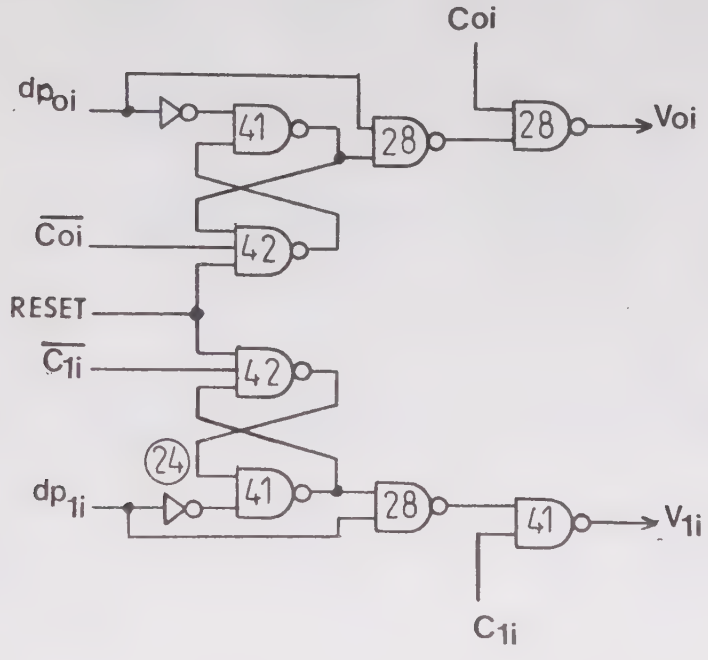

a décalage
ATC
74F158 (16)
74F158 (17)
74F158 (18)
74F158 (19)
74F158 (20)
74F158 (21)
STC
WTC
74S208 (22)
74S208 (23)
74S208 (24)
DB
DV
DR
HDG 0405 (25)
vidéo
HDG 0405 (26)
HDG 0405 (27)
R
V
B
ORGANE VIDÉO



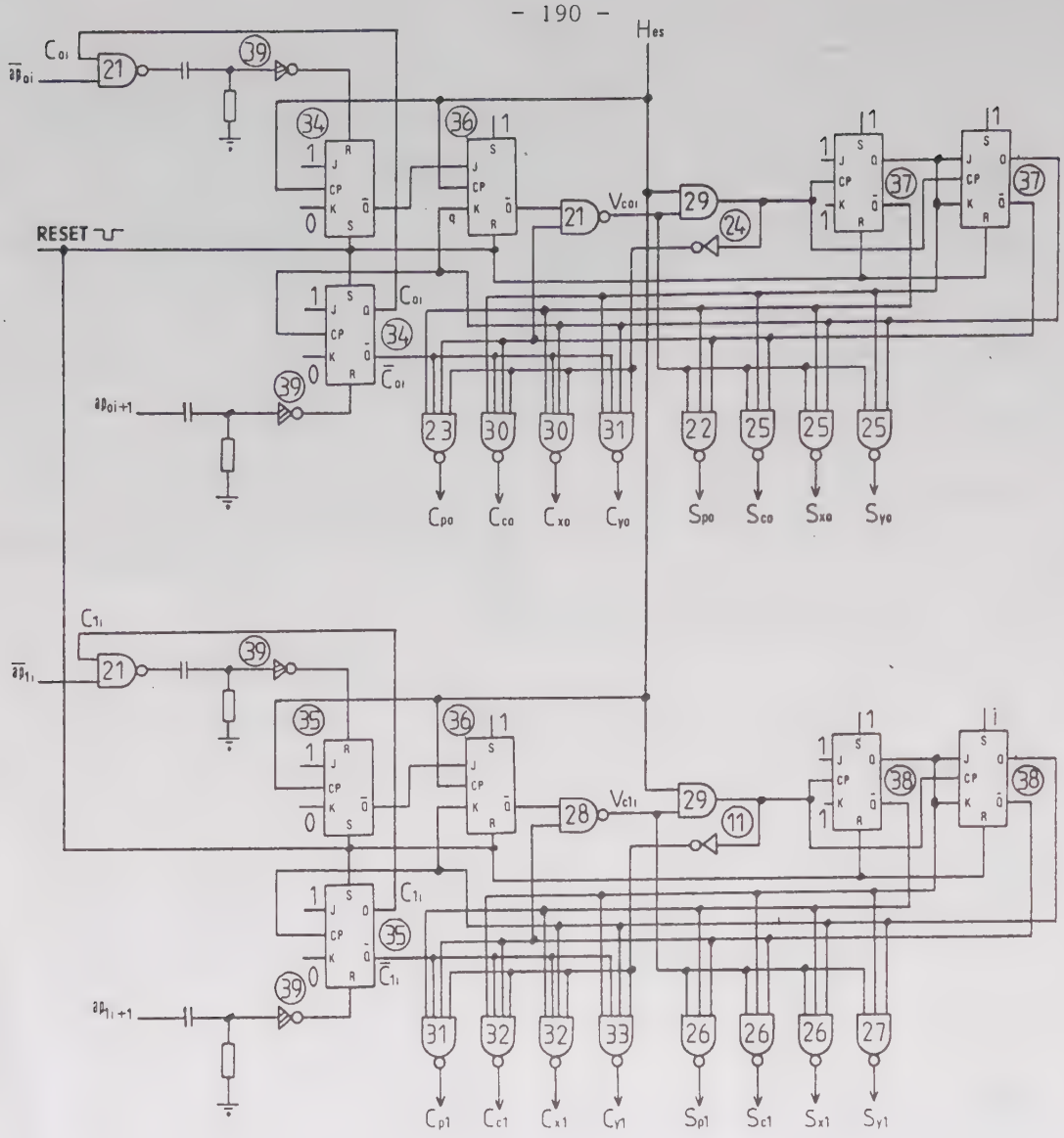
circuit d'autorisation de passage



circuit de demande de passage

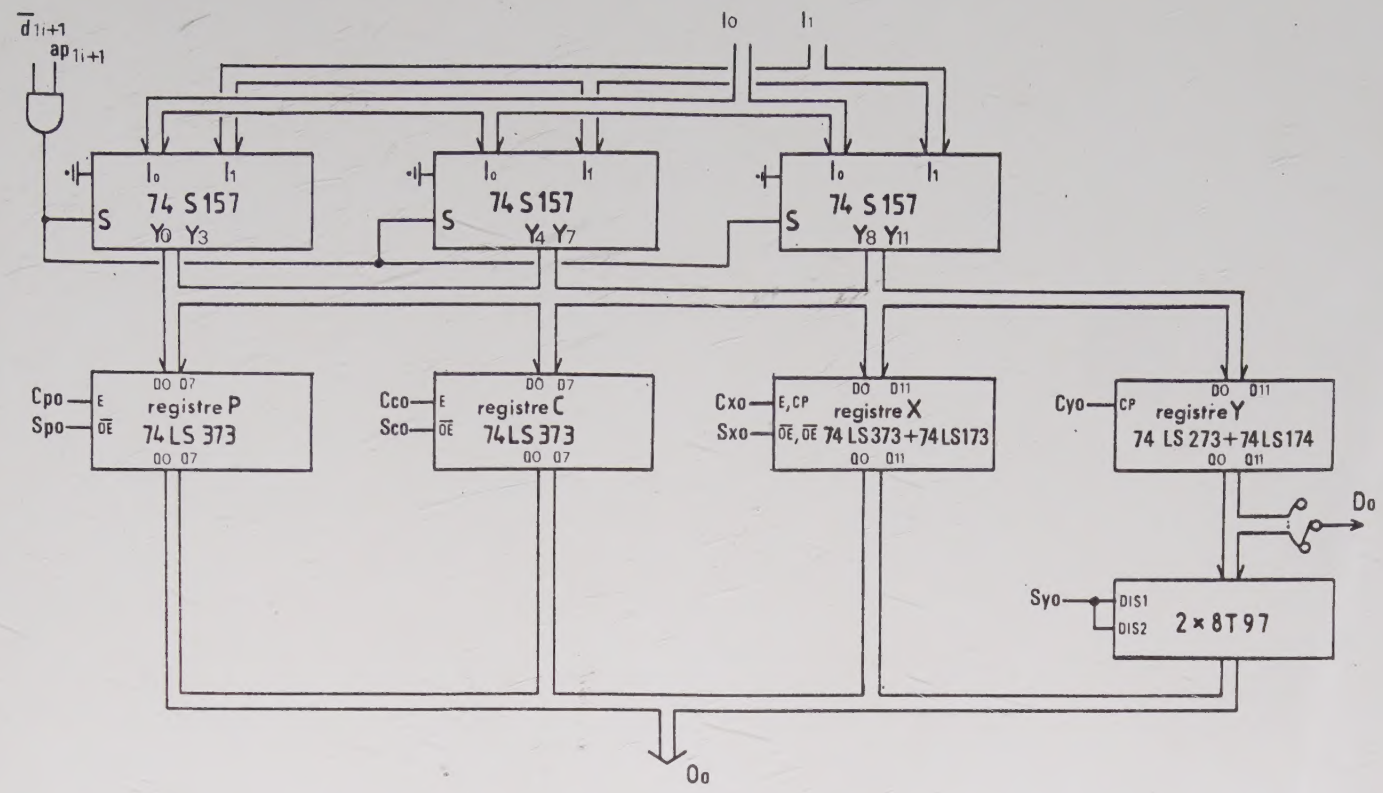


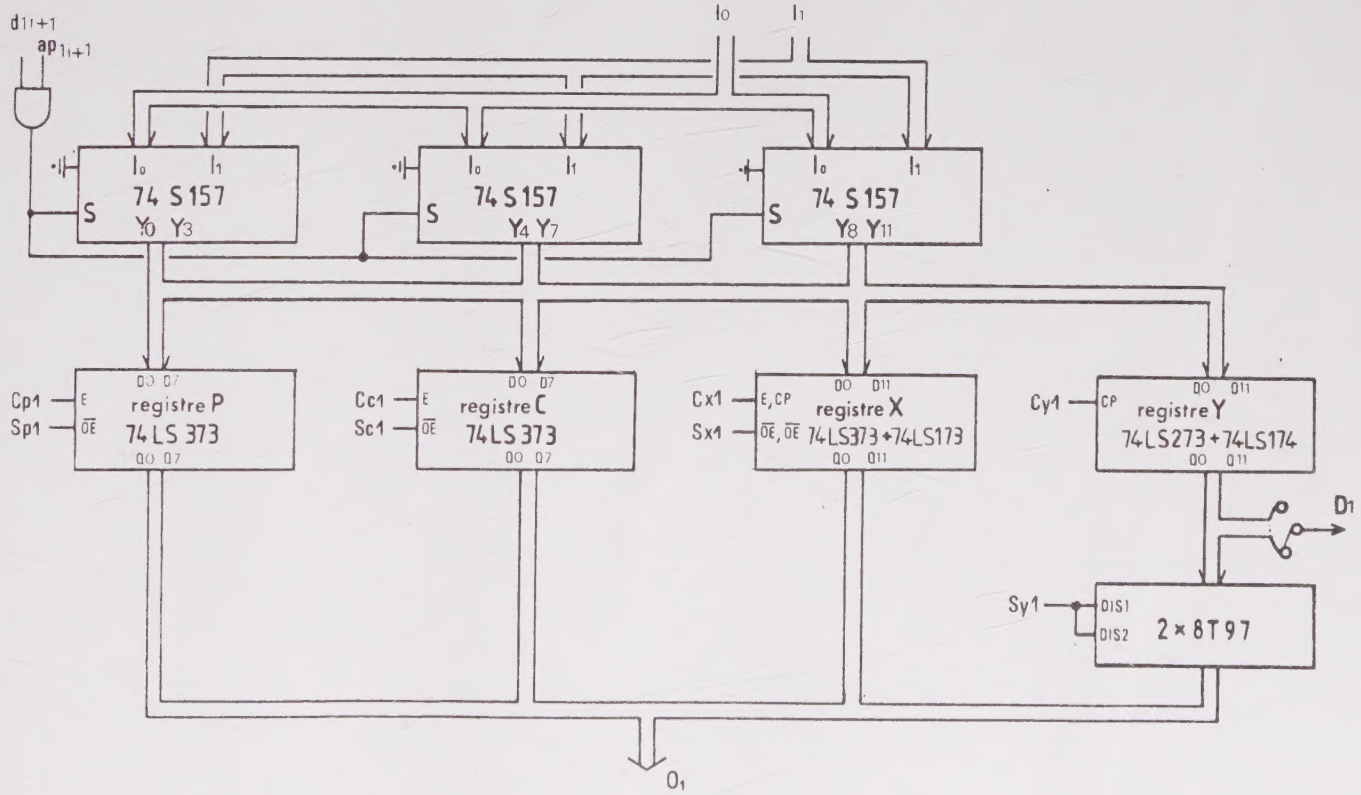
circuit d'état de la voie



circuit de sélection et d'écriture des registres

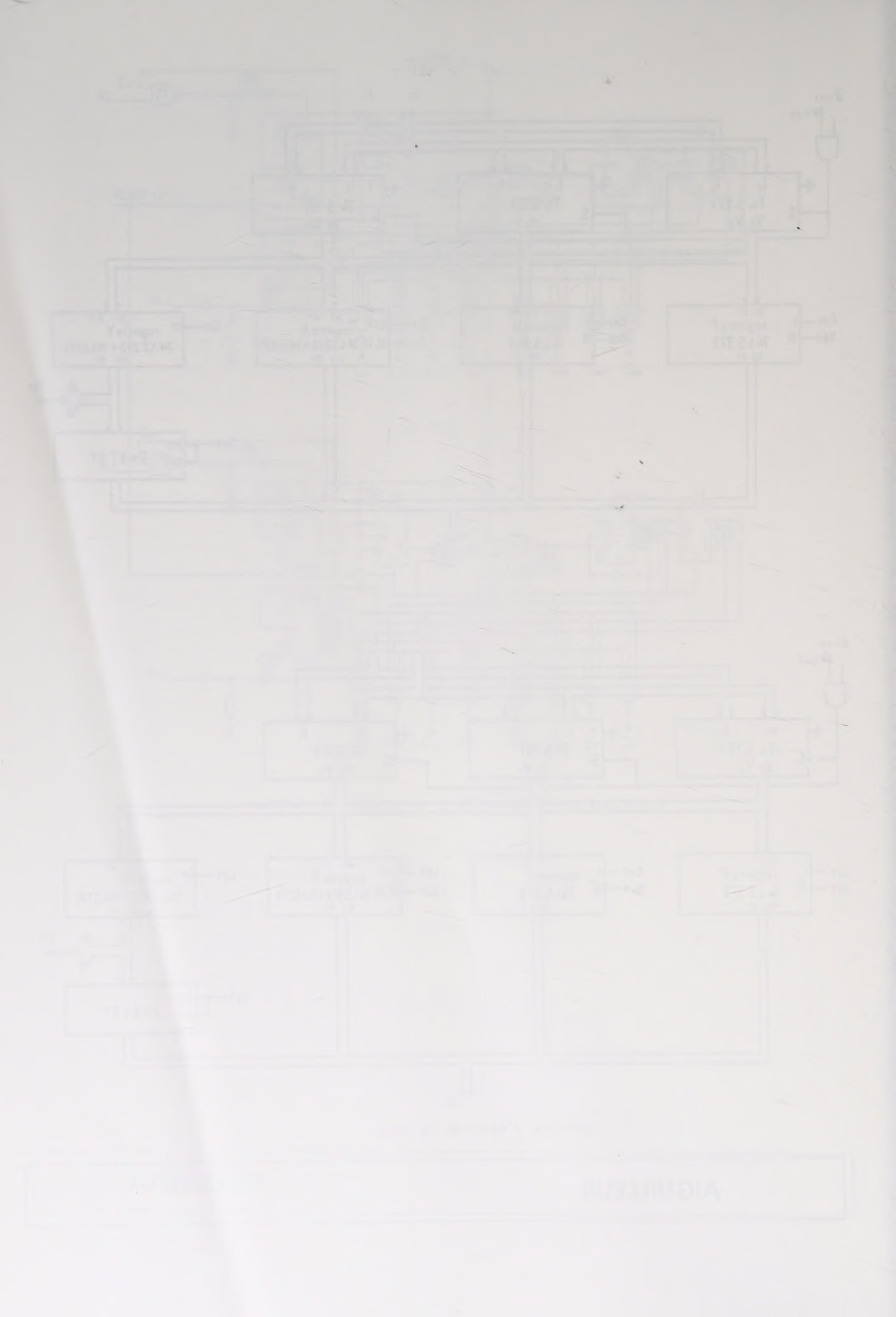
AIGUILLEUR





circuit d'état de la voie

AIGUILLEUR



Résumé:

Cette thèse présente l'architecture multiprocesseurs d'un système graphique à balayage de trame destiné à des applications en CFAO. Le travail porte essentiellement sur l'étude de l'unité d'affichage,dans laquelle est intégrée la fonction calcul des profondeurs,utilisant une mémoire Z.

En vue d'accélérer les algorithmes,ceux-ci sont traités par "n" processeurs travaillant en parallèle. Pour ne pas ralentir ces "n" processeurs et pour avoir la plus grande bande passante au niveau du chargement de la mémoire d'image,celle-ci est divisée en "p" zones indépendantes.

L'interconnexion entre les "n" processeurs et les "p" zones de mémoires utilise un réseau delta du type "$a^n \times b^n$";dans le cas présent, afin d'implémenter les réseaux "$2^k \times 2^r$",il a été développé un aiguilleur "2 x 2".

Mots clés :

architecture multiprocesseurs, système graphique , balayage de trame , unité d'affichage , réseau d'interconnexion , réseau delta , aiguilleur "2 x 2".